TRANSIENT ELECTROCHEMICAL PHENOMENA: APPLICATIONS OF THE KRAMERS-KRONIG RELATIONS AND PHOTOELECTROCHEMICAL ETCHING OF GAAS

By

JOSE MATTHEW RODRIGUEZ ESTEBAN

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1991

ACKNOWLEDGEMENTS

The author would like to thank the following individuals who have been a source of inspiration and support:

- to Professor Mark E. Orazem for his supervision and patience during the course of this project, and for the many opportunities he has given me these past six years to grow as a researcher and as a person;
- to Professors Timothy J. Anderson, Sheng S. Li, Hong H. Lee, and Chang W. Park for their guidance during the course of this work,
- to Professor Timothy J. Anderson, Dr. Arnold Howard, and Dr. Balu Pathangey for the use of the alloying furnace;
- to Professor Augusto Morrone, Materials Science Department, for the use of the Gatan ultrasonic disk cutter;
- to Jennifer Patterson for being a good listener and counselor;
- to Conrad B. Diem for discussing and editing my publications;
- to Oliver C. Moghissi for four good years of sharing apartments and for the impedance data of the copper electrode system;
- to Andy and Rose Jansen for graciously hosting our group parties;
- to Andy Hiser for the impedance data of the metal hydride system;
- to Pankaj Agarwal for his warm laughter and hospitality;
- to Joseph S. Cho and Mark M. Montoya for their friendship;
- to the members of the Electrochemical Engineering research group who assisted in "cutting and pasting"; and,
- to my wife, Lisa T. Bartolome, for her love and constant encouragement.

This research was supported by the National Science Foundation under Grant Number EET–8617057 at the University of Florida.

TABLE OF CONTENTS

page

ACKNOWLEDGEMENTS ii

LIST OF TABLES v

LIST OF FIGURES vi

ABSTRACT xiii

Chapter

1 INTRODUCTION 1

1.1 The Kramers-Kronig Relations 1
1.2 Photodissolution of GaAs 3

2 THE KRAMERS-KRONIG RELATIONS 5

2.1 Definition of Impedance 5
2.2 Relationships Between Impedance Components 7
2.3 Historical Overview 8
2.3.1 Estimation of Polarization Resistance 8
2.3.2 KK-Algorithms for Consistency Tests 10
2.3.3 The KK Controversy 13
2.4 Concept of Proposed Algorithm 15
2.4.1 Mathematical Development 15
2.4.2 Comparison to Previous Algorithms 17
2.5 Application to Synthetic Data 18
2.5.1 Internally Consistent Data Sets 20
2.5.2 Consistent Data Sets With Scatter 22
2.5.3 Invalid Data Sets 33
2.5.4 The Effect of Truncating High-Frequency Data 40
2.6 Application to Experimental Data 40
2.6.1 Experimental Procedure 44
2.6.2 Analysis of Experimental Data 45
2.6.3 Implications for Equivalent Circuit Fitting 62
2.6.4 Numerical Constraints 66

Chapter page

3 PHOTODISSOLUTION OF GaAs 68

3.1 The Semiconductor-Electrolyte Interface 68
3.1.1 Potential Distribution and Photoeffects 70
3.1.2 Thermodynamic and Kinetic Aspects 77
3.1.3 Stabilization of Photocorrosion 81
3.2 Experimental Methods 84
3.2.1 Objective of the Study 84
3.2.2 Rotating Planar Electrodes 86
3.2.3 Electrode Fabrication 91
3.2.4 Experimental Procedure 92
3.3 Results and Discussion 96
3.3.1 Platinum RRDE 96
3.3.2 Hole Injection Experiments 99
3.3.3 Stabilization Experiments 106

4 CONCLUSIONS 126

4.1 Application of the KK-Relations 126
4.2 Photoelectrochemical Etching of GaAs 128

Appendix

A THE KRAMERS-KRONIG ALGORITHM 130

B THE KRAMERS-KRONIG PROGRAM 141

C POTENTIODYNAMIC SCANS FOR RRDE 222

D POTENTIOSTATIC DATA FOR RRDE 235

REFERENCES 248

BIOGRAPHICAL SKETCH 258

LIST OF TABLES

2.1 Fitted parameters for the equivalent circuit of Figure 2.2d with a variable resistor R_1 using LOMFP[16]. The consistent data set is given for $f = 0.0$. The nonstationary systems with various degrees of inconsistency are given by $f = 0.1$, 0.175, 0.25, and 0.5. 39

2.2 Fitting parameters for the equivalent circuit of Figure 2.29 using Data sets A–C of the metal hydride system. The fitting procedure used was LOMFP [16]. 63

2.3 Comparison of the low-frequency asymptotes for the real impedance obtained from the KK-algorithm and the extrapolation of theoretical impedance using the fitting parameters for the circuit shown in Figure 2.29 63

3.1 Experimental values for the geometric parameter $\beta^{2/3} = I_{r,lim}/I_{d,lim}$. . 97

LIST OF FIGURES

2.1 A representation of the algorithm to calculate the low–frequency impedance values for an experimental data set that truncates in the capacitive region. 19

2.2 The electrical circuits employed to generate synthetic impedance data. 21

2.3 Calculated low–frequency impedance for the simple Randles circuit of Figure 2.2a. The truncation frequency ω_{min} is 10^{-2} Hz and $\omega_0 = 10^{-7}$ Hz. 23

2.4 Calculated low–frequency impedance for the Randles circuit with Warburg impedance shown in Figure 2.2b. The truncation frequency is 0.1 Hz. 24

2.5 Calculated low–frequency impedance for the Randles/CPE electrical circuit of Figure 2.2c. This figure illustrates the effect of increasing truncation frequency, while maintaining the value of ω_0 at 10^{-3} Hz. 25

2.6 Calculated low–frequency **real** impedance for the electrical circuit of Figure 2.2d. This figure illustrates the effect of increasing truncation frequency, while maintaining the value of ω_0 at 10^{-4} Hz. 26

2.7 Calculated low–frequency **imaginary** impedance for the electrical circuit of Figure 2.2d. This figure illustrates the effect of increasing truncation frequency, while maintaining the value of ω_0 at 10^{-4} Hz. 27

2.8 Relative residual plots between the data set *good-noise* and the equivalent circuit fit of Figure 2.2d using LOMFP[16]. The random distribution of the residuals indicates a good equivalent circuit fit. 29

2.9 Relative residual plots between the data set *bad-noise* and the equivalent circuit fit of Figure 2.2d using LOMFP[16]. The nonrandom distribution of the real impedance residuals about the analytic impedance suggests a poor equivalent circuit fit. 30

2.10 Calculated low–frequency impedance for *good-noise*: a data set which exhibited randomly distributed relative residuals for the equivalent circuit fit. This data set is shown to satisfy the KK–relations, with $\omega_0 = 10^{-5}$ Hz and $\omega_{min} = 0.1$ Hz. 31

2.11 Calculated low–frequency impedance for *bad-noise*: a data set which exhibited nonrandomly distributed relative residuals for the equivalent circuit fit. The data set is deemed to be inconsistent with the KK–relations, with $\omega_0 = 10^{-5}$ Hz and $\omega_{min} = 0.1$ Hz. 32

2.12 Calculated low–frequency impedance for a simulated nonstationary system in the frequency range 10^{-4} Hz to 0.1 Hz. The electrical circuit of Figure 2.2d was used with a variable resistor R_1 and positive values of "inconsistency factor" f. 34

2.13 Calculated low–frequency impedance for a simulated nonstationary system in the frequency range 10^{-4} Hz to 0.1 Hz. The electrical circuit of Figure 2.2d was used with a variable resistor R_1 and negative values of "inconsistency factors" f. 35

2.14 Calculated low–frequency impedance for a simulated nonstationary system in the frequency range 10^{-4} Hz to 0.1 Hz. The electrical circuit of Figure 2.2d was used with a variable capacitor C_1 and positive values of "inconsistency factor" f. 36

2.15 Calculated low–frequency impedance for a simulated nonstationary system in the frequency range 10^{-4} Hz to 0.1 Hz. The electrical circuit of Figure 2.2d was used with a variable capacitor C_1 and negative values of "inconsistency factor" f. 37

2.16 The effect on inconsistency on the calculated parameters of the equivalent circuit of Figure 2.2d. 41

2.17 The effect of decreasing ω_{max} on the algorithm attributable to the problem of residuals at the high–frequency extreme. 42

2.18 Effect of pH on the open circuit potential of a copper–disk electrode rotating at 1000 RPM in a 0.5N chloride solution, 25°C. 48

2.19 Simulated nonstationary system with variable electrolyte temperature. Impedance data obtained for a copper–disk electrode rotating at 1000 RPM in a 0.5N chloride solution, pH = 9.4. The frequency scan, from 5 kHz down to 0.05 Hz, was performed at the open circuit potential, -330 mV (SCE), with a 10 mV perturbation and a 10 second pause between measurements at each frequency. 49

2.20 Simulated nonstationary system with variable electrolyte pH. Experimental system as in Figure 2.19 except for a constant temperature of 26°C. The initial pH of the solution was 9.4, and the open circuit potential was -215 mV (SCE). 50

2.21 Calculated low–frequency impedance for the nonstationary system of Figure 2.19 with variable solution temperature. 52

2.22 Calculated low–frequency impedance for the nonstationary system of Figure 2.20 with variable solution pH. 53

2.23 Impedance plane plot for the hydrogenation of a metal hydride electrode in 31 wt% KOH solution, −1.10 V (Hg/HgO). 56

2.24 Calculated low-frequency impedance values for Data sets A, B, and C of the metal hydride system. These figures illustrate the results of the KK–algorithm as applied to the entire data set and shows the inconsistency of the available data due to the instability of the electrode. of the system. 57

2.25 Calculated low-frequency impedance values for Data sets A, B, and C of the metal hydride system. These figures show the results of the KK–algorithm after the lower-frequency data are "deleted" to make the remaining data satisfy the KK–relations. 58

2.26 Impedance plane plot for a rotating copper-disk electrode at 1000 RPM in a 0.5N chloride solution at the open circuit potential, pH = 9.4. . . 59

2.27 Calculated low-frequency impedance values for Data sets I, II, and III of the rotating copper-disk system. As in Figure 2.24, the results of the KK–algorithm when applied to the entire data set show the inconsistency of the available data. 60

2.28 Calculated low-frequency impedance values for Data sets I, II, and III of the rotating copper-disk system after deletion of several low–frequency data points. 61

2.29 Equivalent electrical circuit used for the fitting procedure LOMFP [16] for Data sets A–C of the metal hydride system. 64

2.30 Comparison between the calculated impedance values using the KK–algorithm (⊙ and □) and the extrapolated impedance using the fitting parameters of an equivalent circuit fit (solid line), where the symbols ⊙ and □ represent the experimental Data sets A–C of the metal hydride system. 65

3.1 Schematic illustration of a regenerative solar cell for an illuminated n-type semiconductor photoanode in an electrolyte solution containing a redox couple A/A^+. 69

3.2 Schematic illustration of the potential distribution for a semiconductor-electrolyte interface. 72

3.3 Energy band diagram for an illuminated semiconductor-electrolyte interface. 74

3.4 Steady-state photocurrent as a function of disk electrode potential for an n-type Si-doped GaAs disk electrode, 0.5cm diameter, rotating at 1000 RPM. 75

3.5 Saturation photocurrent as a function of light intensity (percent transmission relative to the maximum intensity). 76

3.6 Pourbaix or predominance diagram for GaAs in water by Park and Barber [76]. 79

3.7 A reaction scheme for the decomposition, stabilization, and surface recombination processes for the GaAs/Fe^{2+} photoanode proposed by Vanmaekelbergh *et al.* [85]. 85

3.8 Schematic diagram of the GaAs–disk platinum–ring assembly. 93

3.9 Schematic diagram of the experimental setup. 94

3.10 Limiting currents at the platinum ring (∗) and platinum disk electrode (□) as a function of rotation speed. 98

3.11 Current–potential curves for an n-GaAs disk and platinum ring electrode rotating at 100 RPM in 0.10M KOH and 0.10M $K_3Fe(CN)_6$ solution. 101

3.12 Current–potential curves for an n-GaAs disk and platinum ring electrode rotating at 500 RPM in 0.10M KOH and 0.10M $K_3Fe(CN)_6$ solution. 103

3.13 Photocurrent as a function of relative light intensity (percent transmission) at the n-GaAs disk–Pt ring electrode described in Figure 3.12. 104

3.14 Partial anodic and cathodic current curves at the n-GaAs disk electrode, rotating at 100 RPM, calculated from the I–V curves shown in Figure 3.11. 107

3.15 Partial anodic current curves at the n-GaAs disk electrode, rotating at 500 RPM, calculated from the I–V curves shown in Figure 3.12. . . . 108

3.16 Potentiodynamic scan for an n-GaAs disk Pt-ring electrode under dark conditions rotating at 1000 RPM in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution. The ring electrode was held at a constant potential of −0.2V (Ag/AgCl). 110

3.17 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 500 RPM in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution. The relative light intensity was 50%T. The ring electrode was held at a constant potential of −0.2V (Ag/AgCl). 112

3.18 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 500 RPM in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution. The relative light intensity was 100%T. The ring electrode was held at a constant potential of −0.2V (Ag/AgCl). 113

3.19 Averaged disk and ring currents obtained from Figures C.1 to C.6 for the 50% light transmission as a function of the square root of rotation frequency. 114

3.20 Averaged disk and ring currents obtained from Figures C.7 to C.12 for the 100% light transmission as a function of the square root of rotation frequency. 115

3.21 Disk and ring currents for the potential step experiment (+0.1V (Ag/AgCl)) of an n-GaAs-disk/Pt-ring electrode in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution, with a rotation speed of 500 RPM. The relative light intensity was 20%T, and the potential of the ring electrode was held constant at −0.2V. 118

3.22 Disk and ring currents for the potential step experiment (−0.25V (Ag/AgCl)) using the same experimental system described in Figure 3.21. 119

3.23 Disk and ring currents for the potential step experiment (−0.35V (Ag/AgCl)) using the same experimental system described in Figure 3.21. 120

3.24 Partial disk currents and stabilization ratio for the experimental system described in Figure 3.21 when the disk potential is +0.1V (Ag/AgCl). 123

3.25 Partial disk currents for the experimental system described in Figure 3.21 when the disk potential is −0.35V (Ag/AgCl). 125

C.1 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 500 RPM in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution. The relative light intensity was 50%T. The ring electrode was held at a constant potential of −0.2V (Ag/AgCl). 223

C.2 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 300 RPM and 50% light transmission. 224

C.3 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 500 RPM and 50% light transmission. 225

C.4 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 700 RPM and 50% light transmission. 226

C.5 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 1000 RPM and 50% light transmission. 227

C.6 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 1300 RPM and 50% light transmission. 228

C.7 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 150 RPM in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution. The relative light intensity was 100%T. The ring electrode was held at a constant potential of −0.2V (Ag/AgCl). 229

C.8 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 300 RPM and 100% light transmission. 230

C.9 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 500 RPM and 100% light transmission. 231

C.10 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 700 RPM and 100% light transmission. 232

C.11 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 1000 RPM and 100% light transmission. 233

C.12 Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 1300 RPM and 100% light transmission. 234

D.1 Disk and ring currents for the potential step experiment (+0.1V (Ag/AgCl)) of an n-GaAs-disk/Pt-ring electrode in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution, with a rotation speed of 500 RPM. The relative light intensity was 20%T, and the potential of the ring electrode was held constant at −0.2V. 236

D.2 Disk and ring currents for the potential step experiment (+0.0V (Ag/AgCl)) using the same experimental system described in Figure D.1. 237

D.3 Disk and ring currents for the potential step experiment (−0.1V (Ag/AgCl)) using the same experimental system described in Figure D.1. 238

D.4 Disk and ring currents for the potential step experiment (−0.2V (Ag/AgCl)) using the same experimental system described in Figure D.1. 239

D.5 Disk and ring currents for the potential step experiment (−0.25V (Ag/AgCl)) using the same experimental system described in Figure D.1. 240

D.6 Disk and ring currents for the potential step experiment (−0.3V (Ag/AgCl)) using the same experimental system described in Figure D.1. 241

D.7 Disk and ring currents for the potential step experiment (−0.35V (Ag/AgCl)) using the same experimental system described in Figure D.1. 242

D.8 Partial disk currents and stabilization ratio for the experimental system described in Figure D.1 when the disk potential is +0.1V (Ag/AgCl). 243

D.9 Partial disk currents and stabilization ratio for the experimental system described in Figure D.1 when the disk potential is +0.0V (Ag/AgCl). 244

D.10 Partial disk currents and stabilization ratio for the experimental system described in Figure D.1 when the disk potential is −0.1V (Ag/AgCl). 245

D.11 Partial disk currents and stabilization ratio for the experimental system described in Figure D.1 when the disk potential is −0.2V (Ag/AgCl). 246

D.12 Partial disk currents for the experimental system described in Figure D.1 when the disk potential is −0.35V (Ag/AgCl). 247

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

TRANSIENT ELECTROCHEMICAL PHENOMENA: APPLICATIONS OF THE KRAMERS–KRONIG RELATIONS AND PHOTOELECTROCHEMICAL ETCHING OF GaAs

By

Jose Matthew Rodriguez Esteban

December, 1991

Chairman: Prof. Mark E. Orazem
Major Department: Chemical Engineering

The use of the Kramers–Kronig (KK) relations to test the consistency of impedance data has been limited by the fact that the experimental frequency domain is necessarily finite. Previous numerical algorithms do not distinguish between the residual errors caused by a frequency domain that is too narrow and discrepancies caused by a system which does not satisfy the constraints of the KK–equations. A technique is presented for using the relations to evaluate the consistency of impedance data that do not exhibit a DC limit. The approach presented here differs from previous algorithms because extrapolation of experimental data, required to evaluate the integral relations, is achieved without the assumption of an equivalent circuit and without extrapolation of regressed polynomial equations. The proposed algorithm calculates impedance values below the lowest experimental frequency ω_{min} which "force" the data

set to satisfy the KK–relations. Consistent data sets are shown to yield calculated impedance values which are finite and continuous with the experimental data at ω_{min}. A discontinuity between the calculated low–frequency values and the experimental data indicates inconsistency which cannot be attributed to the finite experimental frequency domain.

The photoelectrochemical etching of n-type GaAs was investigated using the rotating ring–disk electrode. The stabilization of the photodecomposition process by Fe^{2+} as a function of the light intensity and applied potential is presented. At low light intensities, the reducing agent is shown to compete with the corrosion reactions in the potential region of light–limited photocurrent. The stabilization process is shown to be time–dependent due to changing surface characteristics of the GaAs photoanode. Below the potential region of light–limited photocurrent, the rate of the photodecomposition process appears limited initially by the recombination reactions at the semiconductor–electrolyte interface.

CHAPTER 1
INTRODUCTION

Transient electrochemical phenomena are commonly encountered in the study of corrosion systems. The current-voltage characteristics of such systems are inherently time-dependent due to significant, and often irreversible, changes at the substrate surface. The experimental work described here is typical of nonstationary systems, such as corrosion of copper and etching of semiconductors. An algorithm for the Kramers-Kronig relations was developed to address the need to test the consistency of electrochemical impedance data. The algorithm was shown to be a sensitive test for violations of the stability criterion. This work, presented in Chapter 2, resolved some issues discussed in literature concerning the application of these relations. The experimental work for etching GaAs is presented in Chapter 3. The results showed the transient behavior of the photodecomposition reactions at an applied potential. The conclusions are presented in Chapter 4.

1.1 The Kramers-Kronig Relations

Impedance spectroscopy (IS) has become an important technique for the study of electrochemical systems [1, 2], particularly in the area of metal corrosion. This experimental method complements the classical techniques of weight-loss measurements and linear polarization methods (*e.g.,* cyclic voltammetry, chronopotentiometry, and chronoamperommetry). The interpretation of traditional polarization curves is well

understood and provides estimates for kinetic parameters such as the Tafel slopes, corrosion potential, corrosion rate, and solution Ohmic resistance [3–14]. Impedance spectroscopy provides additional information about the metal-electrolyte interface.

Impedance data are typically obtained by applying a sinusoidal potential or current at a single frequency about a steady-state condition and by measuring the resulting current or potential. An impedance spectrum represents the electrical response of an electrochemical system over the widest possible frequency domain. Since elementary processes have different characteristic time constants, the individual processes of a complicated experimental system become dominant at various regions of the frequency domain. "Deconvolution" of the impedance spectrum yields information about the solution Ohmic resistance, double layer capacitance, faradaic resistances, and diffusion limitations.

Two approaches to the analysis and interpretation of impedance data are found in literature. Many workers have used the equivalent electrical circuit (EC) approach where the available data are fitted to the theoretical impedance response of an electrical circuit using nonlinear least-squares regression [2,15–19]. Although an equivalent circuit can be found to adequately model the system, there is often ambiguity about the choice of the circuit and the physical interpretation of the circuit elements. A more rigorous and fundamental approach is the development of transport-based models which solve the governing Laplace's equation and boundary conditions based on the possible reactions at the metal-electrolyte interface, homogeneous reactions, and associated diffusion limitations [1,20–23]. The two approaches commonly assume that the impedance data are characteristic of a linear and stable system.

To facilitate the interpretation of impedance measurements, an investigator should know whether the available experimental data can be used for models that assume linearity and stability. It has been suggested that the Kramers-Kronig (KK) relations

can be employed to evaluate and analyze complex impedance data of electrochemical systems [24–29]. The Kramers–Kronig relations are integral equations which interrelate the frequency dependence of impedance components. In principle, these relations can be used to determine whether the impedance spectrum of a given system has been influenced by time-dependent phenomena. Although this information is critical to the analysis of impedance data, the KK-relations have not found wide-spread use in the analysis of experimental data due to difficulties concerning their application.

A discussion of the KK-relations and the controversy concerning their usefulness for evaluating the consistency of experimental impedance data is presented in Chapter 2. A technique is proposed which employs the KK-relations in a manner that circumvents some concerns discussed in the literature. The concept, mathematical development, and application to synthetic and experimental data will also be discussed. The detailed description of the algorithm is presented in Appendix A. A computer listing for this algorithm is documented in Appendix B.

1.2 Photodissolution of GaAs

The application of semiconductors as photoelectrochemical solar cells is limited by the susceptibility of the substrate to corrode under illumination. The stabilization of photocorrosion processes and the maximization of solar conversion efficiency became major research goals over the past three decades, driven by a desire to develop an inexpensive and long-lived solar cell [30–32]. This work has led to an understanding of the thermodynamic and kinetic aspects of the semiconductor-electrolyte interface. Taking advantage of the instability of semiconductors in electrolyte solutions, laser-induced and masked-light controlled corrosion are currently being employed for the processing of semiconductor devices.

Photoelectrochemical etching and stabilization of GaAs photoanodes were studied using a rotating ring-disk electrode configuration. A review of literature on previous work on semiconductor photoelectrochemical cells is presented in Chapter 3. The first section is a general discussion on the semiconductor-electrolyte interface: potential distribution, thermodynamic aspects, and electrode kinetics. Previous work on the stabilization of photocorrosion is also discussed. The experimental results and discussion are presented in the final section of the chapter.

CHAPTER 2
THE KRAMERS–KRONIG RELATIONS

A discussion of the KK–relations and the controversy concerning their usefulness for evaluating the consistency of experimental impedance data is presented here. The mathematical development and application of the proposed KK–algorithm will be described. The utility of the proposed technique is illustrated using synthetic data derived from equivalent circuits. Since the conditions of the KK–relations are necessarily satisfied, the sensitivity of the technique for checking the violation of the stability criterion was investigated. The procedure was also applied to time–dependent processes such as the corrosion of copper and the hydrogenation of metal hydrides. The technique is shown to be sensitive for testing violations of the stability condition for electrochemical systems and is used to identify a lower frequency bound for data that satisfy the KK–relations. The implications to equivalent circuit fitting procedures and to the estimation of the polarization resistance are discussed. The conclusions are presented in Chapter 4.

2.1 Definition of Impedance

The standard procedure [1, 2] for impedance measurement is to impose on the system a sinusoidal potential

$$V(t) = V_s + V_m \sin(\omega t)$$

with amplitude V_m and frequency ω. The phase lag ϕ and amplitude I_m of the resulting current

$$I(t) = I_s + I_m \sin(\omega t + \phi)$$

are measured. The steady–state potential and current are given by V_s and I_s, respectively. The impedance $Z(\omega)$ relates the potential stimulus to the current response

$$V(t) \;=\; Z(\omega)\, I(t)\,. \tag{2.1}$$

It is a function of frequency such that

$$Z(\omega) \;=\; \mid Z \mid e^{j\phi(\omega)} \tag{2.2}$$

where the modulus $\mid Z \mid = V_m/I_m(\omega)$. Using Euler's relation, this complex quantity can be expressed in terms of its real and imaginary components

$$Z(\omega) \;=\; Z_r(\omega) + j\, Z_i(\omega) \tag{2.3}$$

where

$$Z_r(\omega) \;=\; \mid Z \mid \cos\phi(\omega) \tag{2.4}$$

and

$$Z_i(\omega) \;=\; \mid Z \mid \sin\phi(\omega)\,. \tag{2.5}$$

The impedance response should be characteristic of a system that is causal, linear and stable. The causality condition stipulates that the resulting current is a response to the perturbing potential alone. Furthermore, there should be a linear relationship between the potential and current; this is achieved experimentally by using small amplitude signals. The stability condition specifies that the system should return to its original steady-state condition when the perturbing signal is terminated. This is a controversial constraint, especially for corroding systems where the metal electrode may change during the course of the frequency scan.

2.2 Relationships Between Impedance Components

The Kramers–Kronig relations, developed for the field of optics, are integral equations which constrain the real and imaginary components of complex physical quantities for systems that satisfy the conditions of causality, linearity, stability, and finite impedance values at the frequency limits of zero and infinity [33–37]. Bode [37] extended the concept to electrical impedance, and has derived and tabulated various forms of these equations. Several transformations used in the electrochemical literature are listed below:

$$Z_i(\omega) = -\left(\frac{2\omega}{\pi}\right) \int_0^\infty \frac{Z_r(x) - Z_r(\omega)}{x^2 - w^2}\, dx \,, \tag{2.6}$$

$$Z_r(\omega) = Z_r(\infty) + \frac{2}{\pi} \int_0^\infty \frac{x\, Z_i(x) - \omega\, Z_i(\omega)}{x^2 - w^2}\, dx \,, \tag{2.7}$$

$$Z_r(\omega) = Z_r(0) + \frac{2\omega}{\pi} \int_0^\infty \frac{(\omega/x)\, Z_i(x) - Z_i(\omega)}{x^2 - w^2}\, dx \,, \tag{2.8}$$

and

$$\phi(\omega) = \left(\frac{2\omega}{\pi}\right) \int_0^\infty \frac{\ln \mid Z(x) \mid}{x^2 - w^2}\, dx \,, \tag{2.9}$$

where $Z_r(\omega)$ and $Z_i(\omega)$ are the real and imaginary components of the impedance, ω and x are angular frequencies, $\phi(\omega)$ is the phase lag, and $\mid Z(x) \mid$ is the modulus. These equations imply that there exists a unique transformation for one component if the other is known over the frequency limits of zero to infinity. The usual way of using these equations is to calculate the imaginary component of impedance from the measured real component using equation (2.6) and to compare the values obtained to the experimental imaginary component. Alternatively, the real component of impedance can be calculated from the measured imaginary values using equation (2.7) or (2.8).

Macdonald and Urquidi-Macdonald [38] have demonstrated analytically that equivalent electrical circuits involving passive elements (R, C, L), the Warburg impedance, the pore diffusion model, R–C transmission lines, R–L transmission lines, and R–C–L transmission lines obey the KK–relations (see related discussion in references [39] and [40]). In principle, experimental data that can be fitted to the analytical response of an equivalent circuit described above through nonlinear regression analysis satisfy the conditions of the KK–relations [41].

2.3 Historical Overview

Several numerical algorithms derived from the Kramers–Kronig relations have appeared in the electrochemical literature. These algorithms have been used to estimate the polarization resistance of corrosion systems and to evaluate the consistency of experimental impedance spectra. A discussion of these procedures is presented in this section.

2.3.1 Estimation of Polarization Resistance

The polarization resistance R_p of a freely-corroding system is an important parameter for corrosion engineers. R_p can be estimated from polarization curves and is defined to be the slope of the potential *vs.* current plot at the corrosion potential Φ_{corr}, *i.e.*,

$$R_p = \left(\frac{d\Phi}{dI}\right)_{\Phi_{\text{corr}}} . \tag{2.10}$$

Then corrosion current I_{corr}, and hence the corrosion rate, can be calculated from the Stern-Geary equation [3, 7]

$$I_{\text{corr}} = \frac{B_a\, B_c}{2.303\,(B_a + B_c)} \cdot \frac{1}{R_p} \tag{2.11}$$

if R_p and the anodic and cathodic Tafel slopes (B_a and B_c, respectively) are known.

Impedance spectroscopy has also been used to estimate the polarization resistance

$$\begin{aligned} R_p &= \mid Z \mid_{\omega \to \infty} - \mid Z \mid_{\omega \to 0} \\ &= Z_r(\infty) - Z_r(0) . \end{aligned} \tag{2.12}$$

Using the KK–relations (2.7) and (2.8), equation (2.12) can be rewritten as

$$R_p = Z_r(\infty) - Z_r(0) = \frac{2}{\pi} \int_0^\infty \left[\frac{Z_i(x)}{x} \right] dx . \tag{2.13}$$

Therefore, R_p can be obtained from either the real component or imaginary component alone if the available impedance spectrum extends over a sufficiently wide frequency domain. Equation (2.12) provides an estimate of R_p and corrosion rate from the frequency limits of the real component, while equation (2.13) can also be used to evaluate the consistency of impedance data using the imaginary component.

Kendig and Mansfeld [42] have used the KK–relations to estimate the R_p for electrochemical systems that exhibit nearly symmetrical impedance spectra. They proposed evaluating the integral of equation (2.13) using only the upper half of the frequency domain wherein the imaginary component exhibits a maximum value. Their approach severely limits the utility of equation (2.13) to systems which can be modeled by a single relaxation time constant [26, 43]. Impedance spectra found in literature are rarely symmetric in the frequency domain due to multiple elementary processes which have different characteristic time constants.

Macdonald and Urquidi-Macdonald [26] proposed a more general algorithm which evaluates the integral of equation (2.13) over the entire experimental frequency domain. They showed that estimated values of R_p using equation (2.12) and (2.13) are comparable if the impedance spectrum spans a sufficiently broad frequency domain. In their example of a rapidly corroding system, the estimation for R_p was seriously affected because the data were less extensive and of lower quality. Nevertheless, they stated that the transformation was satisfactory even when fulfillment of the stability condition was questionable.

The application of equation (2.13) requires impedance data over an infinite frequency bandwidth. Experimental data, however, are constrained within a finite frequency domain. The impedance spectrum may not even exhibit a DC limit at low frequencies such as for highly corrosion-resistant metals (see examples in Shih and Mansfeld [17, 18]). In addition, the instability of the metal electrode due to corrosion results in excessive noise during impedance measurements at low frequencies. It can be easily seen that the impedance in the low-frequency domain contributes significantly to the integral of equation (2.13). Consequently, this relation alone is not a reliable check for the consistency of experimental data especially when a significant portion of the impedance spectra in the low-frequency domain is unmeasured. The algorithms described above will not discriminate between the errors caused by a frequency domain that is too narrow and discrepancies caused by a system which does not satisfy the constraints of the KK–equations.

2.3.2 KK–Algorithms for Consistency Tests

Macdonald and Urquidi-Macdonald [26] recognized the inadequacy of using equation (2.13) alone as a consistency test and proposed other algorithms which used the KK–relations given by equations (2.6)–(2.8). These integral equations were rewritten in what were referred to as "finite difference forms" [27]. The algorithms essentially employed Simpson's Rule to numerically integrate the KK–relations. The experimental frequency domain was divided into several segments, and the discrete impedance components $Z_r(\omega)$ and $Z_i(\omega)$ were fitted to a polynomial expression given by

$$Z_{r \text{ or } i} = \sum_{k=0}^{n} a_k \omega^k \tag{2.14}$$

In addition, they extrapolated the impedance spectrum beyond the experimental frequency domain using polynomial expressions derived from a least-squares fit of the experimental data. Although this approach is questionable, they stated that the

procedure was meant to minimize the problem of residuals associated with the unmeasured impedance in the inaccessible frequency regions. They illustrated the usefulness of the algorithms to test the consistency of synthetic impedance data derived from equivalent electrical circuits, which necessarily satisfy the KK–relations, and of a stable experimental system, TiO_2-coated carbon steel in HCl/KCl solution. They also reported that the "accuracy of the transforms is sensitive to noise and instabilities inherent in a rapidly-corroding interface, particularly in the electrodissolution and open-circuit potential regions." In a subsequent paper, Urquidi–Macdonald *et al.* [28] reported that the algorithms they proposed were relatively insensitive to violations of the linearity condition but were sensitive to violations of the stability criterion. They explained that modern frequency response analyzers reject harmonics of the current response which manifest nonlinear behavior thereby effectively "linearizing" the response. Such anomalies can be observed by varying the amplitude of the voltage perturbation. Violation of the stability condition was tested for an iron–H_2SO_4 system by applying a linear voltage sweep on the electrode interface while simultaneously measuring the impedance. The applied voltage had changed by almost one volt over the span of two hours. Although this may be a severe test, they argued that the KK–relations were shown to be sensitive to violations of the stability condition even at small changes in applied voltage.

Haili [29] studied the asymptotic behavior of the KK-relations and showed the importance of extrapolating impedance data to both zero and infinite frequency when evaluating the integrals of the relations. For the experimental data in the frequency domain $\omega_{min} \leq \omega \leq \omega_{max}$, the interpolating functions are given by

$$Z_r(\omega) = B_r \omega^{A_r} \tag{2.15}$$

$$-\omega Z_i(\omega) = A_i \ln \omega + B_i \tag{2.16}$$

for the real and imaginary components, respectively. The coefficients A_i, B_i, A_r, and

B_r are coefficients determined for each interval between data points. In contrast to the approach of Macdonald *et al.*, Haili's extrapolating functions were based on the expected asymptotic behavior of a typical electrochemical system. For extrapolation to $\omega = 0$, the imaginary component $Z_i(\omega)$ is proportional to ω as $\omega \rightarrow 0$ if the system behaves as a Randles-type equivalent circuit. The Warburg impedance, which could dominate in the lower frequency region, is also nearly proportional to ω as $\omega \rightarrow 0$. The real impedance approaches a constant limit which is the sum of the Ohmic solution resistance and the polarization resistance. The extrapolating functions which describe these asymptotic behaviors as $\omega \rightarrow 0$ are given by

$$\omega Z_i(\omega) = A_{I0} w^2 \qquad \text{for } 0 \leq \omega \leq \omega_{min} \tag{2.17}$$

for the imaginary component, with

$$A_{I0} = \frac{Z_i(\omega_{min})}{\omega_{min}}, \tag{2.18}$$

and for the real component

$$Z_r(\omega) = A_{R0}\omega^2 + R_0 \qquad \text{for } 0 \leq \omega \leq \omega_{min} \tag{2.19}$$

with

$$A_{R0} = \frac{Z_r(\omega_{min}) - R_0}{\omega_{min}^2}. \tag{2.20}$$

The extrapolation in this region involves only one adjustable parameter R_0, whose value will approximate $Z_r(\omega_{min})$ if ω_{min} extends to a low enough frequency. At high frequencies, the system exhibits a capacitive behavior due to the double layer capacity and the imaginary component is inversely proportional to frequency as $\omega \rightarrow \infty$. Although $\omega Z_I(\omega)$ increases rapidly at high frequencies, Haili notes that

$$\lim_{\omega \rightarrow \infty} \omega Z_i(\omega) = C \text{ (a constant)}.$$

The real component attains a constant limit as $\omega \rightarrow \infty$ which is equivalent to the Ohmic solution resistance R_∞. These asymptotic behaviors as $\omega \rightarrow \infty$ are given by

$$-\omega Z_i(\omega) = A_{I\infty}\omega + B_{I\infty} \quad \text{for } \omega_{max} \leq \omega \leq \omega_\infty \tag{2.21}$$

$$= C \quad \text{for } \omega > \omega_\infty \tag{2.22}$$

for the imaginary component, with

$$C = A_{I\infty}\omega_\infty + B_{I\infty} \tag{2.23}$$

and for the real component

$$Z_r(\omega) = A_{R\infty}\omega + B_{R\infty} \quad \text{for } \omega_{max} \leq \omega \leq \omega_\infty \tag{2.24}$$

$$= R_\infty \quad \text{for } \omega > \omega_\infty \tag{2.25}$$

where

$$A_{R\infty} = \frac{R_\infty - Z_r(\omega_{max})}{\omega_\infty - \omega_{max}} \quad \text{and} \tag{2.26}$$

$$B_{R\infty} = Z_r(\omega_{max}) - A_{R\infty}\omega_{max}\,. \tag{2.27}$$

The adjustable parameters are the linear coefficients $A_{I\infty}$, $B_{I\infty}$, $A_{R\infty}$, $B_{R\infty}$, and the frequency ω_∞ where $\omega Z_i(\omega)$ and $Z_r(\omega)$ attain constant values. Assuming reasonable values for the listed adjustable parameters, these extrapolating functions were included when evaluating the integrals of equations (2.6) and (2.7). The various examples Haili presented illustrated the importance of accounting for the contribution to the integrals by the unmeasured impedance.

2.3.3 The KK Controversy

There is controversy over the extent to which the KK–relations can be used to validate electrochemical impedance data. These integral relations involve frequencies

ranging from zero to infinity, but experimental data are available only within a finite frequency domain. Macdonald and Urquidi-Macdonald [26] and Urquidi-Macdonald *et al.* [27] have emphasized the importance of acquiring data over the widest possible frequency domain in order to minimize the integration errors associated with the inaccessible frequency "tails," but this is not always possible. While the upper limit of modern EIS instrumentation (65 kHz to 1 MHz) is sufficient for most electrochemical systems, data acquisition at low frequencies is often governed by noise attributable to the instability of the electrode. It is this value that currently restricts the utility of the KK–transforms.

The controversy surrounding the application of the KK–relations to experimental data concerns the proper treatment of the integrals from the inaccessible frequency domain. The relationships are held to be valid, but Shih and Mansfeld [18,44–46] and Mansfeld and Shih [47] have argued that they are useless for data that do not include all the time constants for the system. They stated that Macdonald's algorithms yield erroneous conclusions about the validity of impedance data which do not exhibit a "DC limit." Urquidi-Macdonald *et al.* [28] and Macdonald and Urquidi-Macdonald [48, 49] explained that the origin of Mansfeld and Shih's claim is a failure to recognize the "tails" problem which resulted in the incomplete evaluation of the transform integrals.

The two algorithms described above attempt to resolve the problem using extrapolating functions. Although the approach employed by Urquidi-Macdonald *et al.* [27] reduces the problem of the inaccessible "tails" of the integrals, extrapolation of higher order polynomials in this manner is justifiable only within a narrow frequency domain. The approach used by Haili [29] was more justified because the extrapolation is based upon the asymptotic behavior of typical electrochemical systems. Nevertheless, such procedures may not always be the prudent solution to the problem. The

impedance spectra of highly corrosion-resistant systems usually terminate well within the capacitive region. For such systems, the unmeasured impedance below the lowest experimental frequency are not merely part of a "tail" but represent a significant portion of the impedance spectrum. The accuracy of evaluating the transform integrals will be affected by the choice of an extrapolation procedure.

2.4 Concept of Proposed Algorithm

It is evident from the discussions of Shih and Mansfeld, Macdonald and Urquidi-Macdonald, and Haili that the application of the KK–relations to a broad range of experimental impedance data requires a method to eliminate errors associated with the unmeasured frequency domain. The object of this work was to develop such an algorithm that can distinguish between errors due to a finite frequency domain that is too narrow and discrepancies due to data that are truly inconsistent with the KK–relations. This algorithm is intended to reduce the possibility for false rejection of a valid data set caused by an insufficient experimental frequency domain.

This section outlines the mathematical development for an algorithm which is different from current applications of the KK–relations because it does not require extrapolation of available data following Macdonald's approach or by using the fitting parameters of an equivalent electrical circuit.

2.4.1 Mathematical Development

An algorithm has been developed [50, 51] based on two forms of the KK–relations given by equations (2.6) and (2.7). Through these relations, the value of one impedance component (real or imaginary) can be calculated at any given frequency if the other component is known over the entire frequency range. Experimental

difficulties, however, constrain the acquisition of data within a finite frequency domain $\omega_{min} \leq \omega \leq \omega_{max}$. The apparent lack of agreement between the experimental data and the corresponding KK–transformations can be attributed to two factors: (1) the problem of residuals due to the unmeasured impedance values in the frequency ranges $\omega < \omega_{min}$ and $\omega > \omega_{max}$, and, (2) to processes associated with a nonstationary, nonlinear, and/or noncausal system. The contributions to the integrals of the KK–relations arise predominantly in the region near the frequency being evaluated [25, 29], that is, within two or three orders of magnitude from the frequency being considered. This means that the problem of residuals described by factor (1) will be much more significant at the low frequency range than at the high frequency range if an experimental data set truncates in the capacitive region. This is evident from the examples given by Shih and Mansfeld [44–46] where the greatest discrepancies between the experimental data and the calculated impedance values occurred at low frequencies.

The concept behind this work is that both equations can be used simultaneously to calculate the impedance below the lowest measured frequency ω_{min}. The calculated impedance, in the domain $\omega_0 \leq \omega < \omega_{min}$, "forces" the experimental data set to satisfy the KK–relations in the frequency domain $\omega_0 \leq \omega \leq \omega_{max}$. The low-frequency limit ω_0 is an adjustable parameter that is typically three or four orders of magnitude smaller than ω_{min}. The parameter ω_0 is chosen to satisfy the requirement that the impedance spectrum does not terminate in the capacitive region; *i.e.,* the real component attains an asymptotic value and the imaginary component approaches zero as $\omega \rightarrow \omega_0$. Internal consistency between the impedance components also requires that the calculated functions be continuous with the experimental data at ω_{min}. These requirements cannot simultaneously be satisfied by data from systems that do not satisfy the constraints of the KK–relations. Discontinuities at ω_{min} can

therefore be attributed to properties unrelated to the use of a finite frequency range in the collection of data. A detailed discussion of the mathematical development is given in Appendix A and a complete listing of the computer program is presented in Appendix B.

A major assumption in using this approach is that the highest measured frequency ω_{max} is large enough to ensure that the high-frequency data do not truncate in the capacitive region. Residual errors at the high–frequency extreme do not influence the calculations because the major contribution to the integrals of the KK–relations is from the lower frequency domain. The consequence of this assumption will be discussed in a later section. This assumption can be relaxed by employing polynomial expressions applicable for the high-frequency domain as given in the equations listed in Appendix A.

2.4.2 Comparison to Previous Algorithms

The algorithms proposed by Macdonald and Urquidi-Macdonald [26] Urquidi-Macdonald *et al.* [27], and Haili [29] employed the KK–relations in a "classical" manner as described earlier on page 7. The approach presented here is different from these algorithms because the KK–relations are themselves used to extrapolate data to frequencies below the lowest measured frequency. This does not require extrapolation of the experimental data below the lowest measured frequency using polynomial functions [26, 27, 29] or through the fitting parameters of an equivalent circuit [45].

The object of the algorithm is to determine the impedance in the low frequency domain which, when appended to the experimental data, *will force* the experimental data to satisfy the KK–relations. The extrapolating functions in the low frequency domain $\omega_0 \leq \omega_l < \omega_{min}$ are given by

$$Z_r(\omega_l) = \sum_{k=0}^{K} a_k^{(1)} (\log \omega_l)^k \tag{2.28}$$

$$Z_i(\omega_l) = \sum_{m=0}^{M} a_m^{(2)} (\log \omega_l)^m . \tag{2.29}$$

These functions are polynomial expressions of order K and M, with coefficients $a_0^{(1)}$ to $a_K^{(1)}$ and $a_0^{(2)}$ to $a_M^{(2)}$, respectively. The semi-logarithmic form was chosen because it is well-behaved in the high frequency regions where high-order polynomials [27] (*i.e.*, $Z(\omega) = \sum_{i=0}^{n} a_i \omega^i$) exhibit oscillations. Although both polynomial expressions are well behaved at lower frequencies, the semi-logarithmic form can be used over the entire frequency domain. Equations (2.28) and (2.29) cannot be justified as asymptotic behaviors when $\omega \rightarrow 0$ as compared to the extrapolating functions used by Haili [29]. However, if Haili's extrapolating functions were employed, the asymptotic behavior would restrict the imaginary component to a linear dependence with frequency. It will be shown later that this is not the case for inconsistent impedance data. Equations (2.28) and (2.29) are less restrictive functions, and their "asymptotic" behavior below ω_{min} will be an indication of the consistency of experimental data. A schematic representation of the algorithm is given in Figure 2.1.

2.5 Application to Synthetic Data

To test the approach, the algorithm was applied to several synthetic data sets derived from equivalent electrical circuits. The objectives are listed below.

- *It will be shown that consistent data truncated in the capacitive region will be deemed to be consistent.* Internally consistent data were generated within finite frequency ranges for the various equivalent circuits illustrated in Figure 2.2. The impedance spectra of these equivalent circuits necessarily satisfy the KK–relations; therefore, the ability of the algorithm to calculate the low–frequency

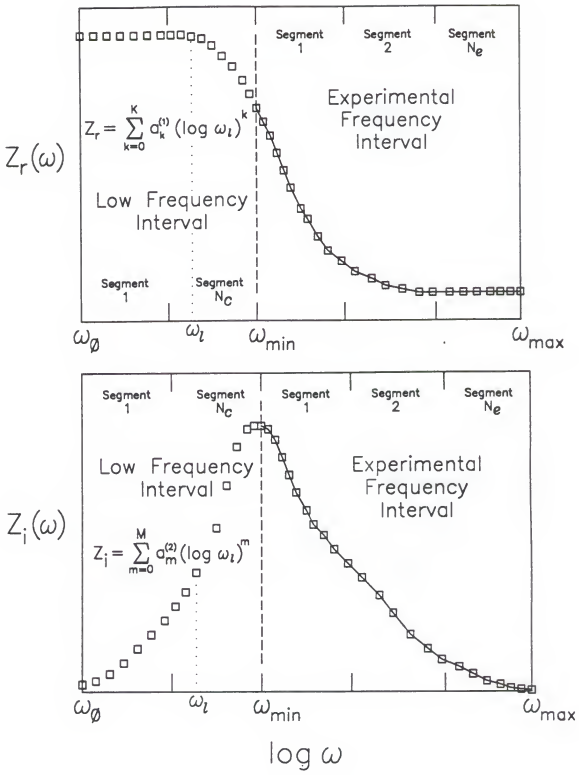


Figure 2.1: A representation of the algorithm to calculate the low–frequency impedance values for an experimental data set that truncates in the capacitive region. The mathematical development of the algorithm is given in Appendix A. Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

impedance values which are continuous with the experimental data at ω_{min} is demonstrated. The analytic impedance equations for the electrical circuits are listed in Appendix A.3.

- *It will be shown that the method can discriminate between errors due to a finite frequency range that is too narrow and discrepancies due to data that are truly inconsistent with the KK–relations.* A nonstationary system was simulated by changing the value of a circuit element through the course of the "experiment." The appearance of a discontinuity at ω_{min} is shown to be characteristic of inconsistent impedance data.

2.5.1 Internally Consistent Data Sets

Synthetic data were generated for the equivalent circuits shown in Figure 2.2. Ten "experimental" data points were obtained per logarithmic decade of frequency. The truncation frequencies were chosen such that the data sets did not exhibit a DC limit in the low frequency range.

The impedance values below ω_{min} were calculated using the KK–algorithm described earlier, and the results are shown in Figures 2.3 to 2.7; the analytic impedance of the electrical circuits are shown for comparison. These figures show that the calculated low–frequency impedance functions, $Z_r(\omega_l)$ and $Z_i(\omega_l)$, are continuous with the experimental data at ω_{min} and closely approximate the theoretical values. The truncation frequencies of the data sets in Figures 2.3 to 2.5 were sufficiently low to provide enough information on all the time constants of the equivalent circuits. The proposed KK–algorithm estimated the lower–frequency impedance to within 5% of the theoretical values when the experimental data provided enough information about the dominant time constant in the lower-frequency domain. It can be shown that the

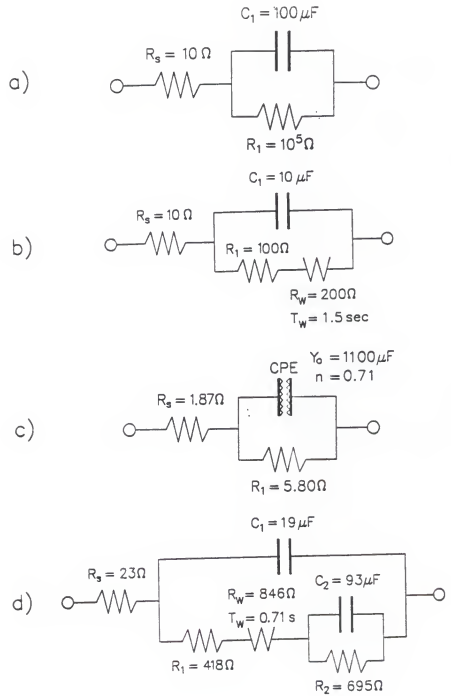


Figure 2.2: The electrical circuits employed to generate synthetic impedance data. The analytic expressions for the impedance components are presented in Appendix A.3. Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

"complete" impedance data sets (*i.e.,* the calculated low frequency impedance values appended to the experimental data set) satisfy the KK–relations using the algorithms suggested by Urquidi-Macdonald *et al.* [27] without further extrapolation. The effect of increasing ω_{min} is illustrated in Figures 2.6 and 2.7. The low frequency impedance values could not be accurately calculated as the experimental data moved further into the capacitive region, as shown in curves **c** and **d**. This reflects the lack of information about the impedance behavior of the system, particularly of the dominant time constant within the low–frequency region. It has been observed for consistent data sets that the continuity of the calculated *real* impedance at ω_{min} is an indication of consistency, while the appearance of a discontinuity for the calculated ***imaginary*** impedance reflects the missing time constant(s) in the lower frequency domain (see, for example, curve **c** of Figures 2.6 and 2.7).

The utility of the approach described here is that an equivalent circuit fit is not required to determine the consistency of the impedance data. Moreover, for data sets which do not exhibit a DC limit, employment of the KK–relations no longer requires extrapolation below the lowest measured frequency using the fitting parameters of an equivalent circuit [45, 46] or extrapolating functions [27, 29]. It should be pointed out that the data set obtained from the electrical circuit of Figure 2.2a was previously employed [45, 46] to illustrate the shortcomings of the KK–algorithm suggested by Urquidi-Macdonald *et al.* [27]. This same data set is deemed to be consistent using the KK–algorithm presented here, as shown in Figure 2.3.

2.5.2 Consistent Data Sets With Scatter

The applicability of the algorithm to data sets with scatter is presented in this section. Two "noisy" data sets (hereafter referred to as *"good-noise"* and *"bad-noise"*) were simulated using the analytic impedance response of the equivalent circuit of

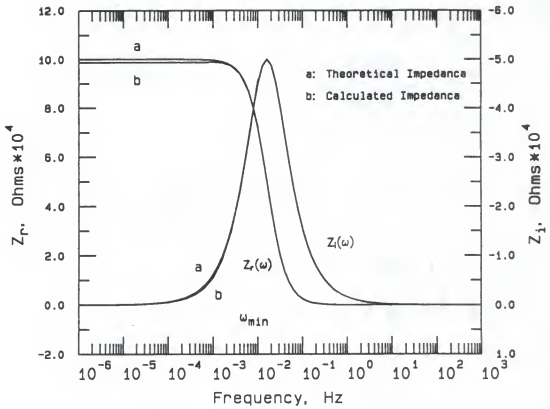


Figure 2.3: Calculated low-frequency impedance for the simple Randles circuit of Figure 2.2a. The truncation frequency ω_{min} is 10^{-2} Hz and $\omega_0 = 10^{-7}$ Hz. a) Theoretical impedance response; and, b) Calculated impedance. Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

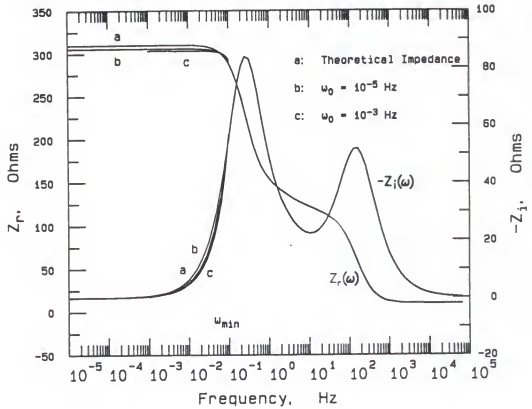


Figure 2.4: Calculated low-frequency impedance for the Randles circuit with Warburg impedance shown in Figure 2.2b. The truncation frequency is 0.1 Hz. The accuracy of the algorithm was improved for Curve b because the problem of residuals was minimized by using a lower value for ω_0. **a)** Theoretical impedance response; **b)** Calculated impedance, $\omega_0 = 10^{-5}$ Hz; and, **c)** Calculated impedance, $\omega_0 = 10^{-3}$ Hz. Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

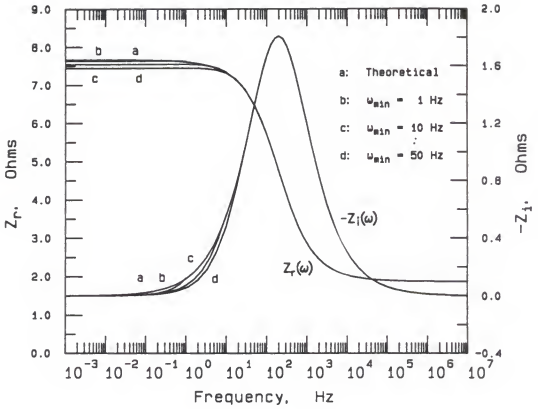


Figure 2.5: Calculated low-frequency impedance for the Randles/CPE electrical circuit of Figure 2.2c. This figure illustrates the effect of increasing truncation frequency, while maintaining the value of ω_0 at 10^{-3} Hz. **a)** Theoretical impedance response; **b)** Calculated impedance, $\omega_{min} = 1$ Hz; **c)** Calculated impedance, $\omega_{min} = 10$ Hz; and, **d)** Calculated impedance, $\omega_{min} = 50$ Hz. Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

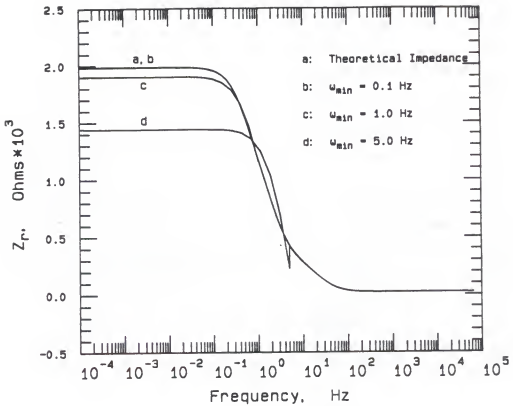


Figure 2.6: Calculated low-frequency **real** impedance for the electrical circuit of Figure 2.2d. This figure illustrates the effect of increasing truncation frequency, while maintaining the value of ω_0 at 10^{-4} Hz. **a)** Theoretical impedance response; **b)** Calculated impedance, $\omega_{min} = 0.1$ Hz; **c)** Calculated impedance, $\omega_{min} = 1$ Hz; **and, d)** Calculated impedance, $\omega_{min} = 5$ Hz. Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

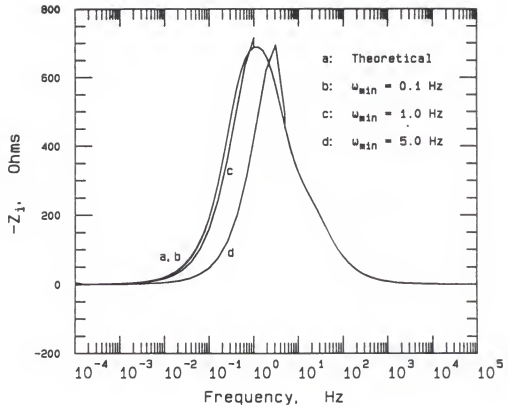


Figure 2.7: Calculated low-frequency **imaginary** impedance for the electrical circuit of Figure 2.2d. This figure illustrates the effect of increasing truncation frequency, while maintaining the value of ω_0 at 10^{-4} Hz. **a)** Theoretical impedance response; **b)** Calculated impedance, $\omega_{min} = 0.1$ Hz; **c)** Calculated impedance, $\omega_{min} = 1$ Hz; and, **d)** Calculated impedance, $\omega_{min} = 5$ Hz. Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

Figure 2.2d and a random number generator. Random "noise" was added or subtracted up to 5% of the impedance value at a given frequency. These data sets were fitted to the equivalent circuit using LOMFP [16], a complex nonlinear least–squares immitance fitting program. The goodness of fit between the experimental data and the analytic response can be visually ascertained from the relative residual plots shown in Figures 2.8 and 2.9, for *good-noise* and *bad-noise*, respectively. A "good" fit will exhibit a random distribution for the residuals about the analytic impedance values, such as that shown in Figure 2.8, and verifies the consistency of the data set called *good-noise*. The residual plots for *bad-noise* displays a nonrandom distribution for the real impedance, and provides a measure of the ill–fit of the data set to the chosen equivalent circuit.

The low–frequency impedance values below ω_{min} were calculated using the KK–algorithm, and the results are presented in Figures 2.10 and 2.11 for *good-noise* and *bad-noise*, respectively. The calculated lower frequency impedance values for *good-noise* are shown to be continuous with the experimental data at the truncation frequency ω_{min} , and is an indication of the internal consistency of the data set. The polynomial expressions defined in equations (2.28) and (2.29) effectively smoothed the scattered impedance data contained in *good-noise*. The results for *bad-noise* show the calculated impedance data below the truncation frequency to be discontinuous with the experimental data at ω_{min} , and suggests that *bad-noise* does not satisfy the KK–relations. The conclusions about the validity of these "noisy" data sets obtained from the KK–algorithm are consistent with the residual–plot analyses derived from LOMFP.

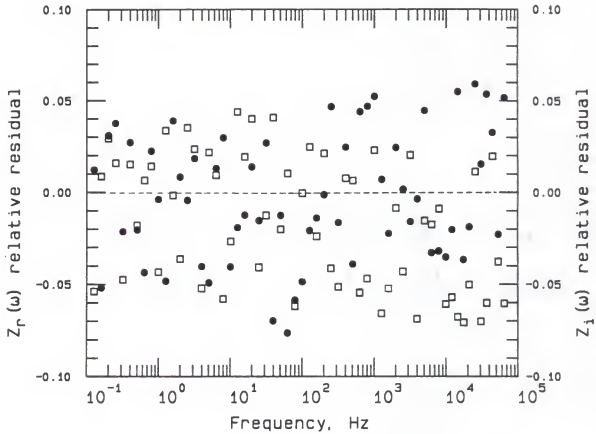


Figure 2.8: Relative residual plots between the data set *good-noise* and the equivalent circuit fit of Figure 2.2d using LOMFP[16]. The random distribution of the residuals indicates a good equivalent circuit fit. (•) Relative residuals for the real impedance; and, (□) Relative residuals for the imaginary impedance. Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

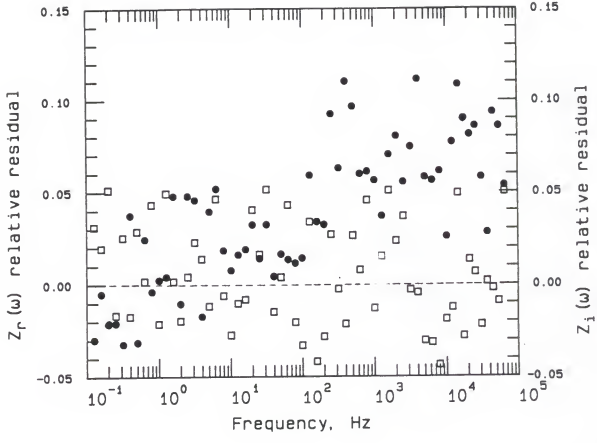


Figure 2.9: Relative residual plots between the data set *bad-noise* and the equivalent circuit fit of Figure 2.2d using LOMFP[16]. The nonrandom distribution of the real impedance residuals about the analytic impedance suggests a poor equivalent circuit fit. (•) Relative residuals for the real impedance; and, (□) Relative residuals for the imaginary impedance. Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

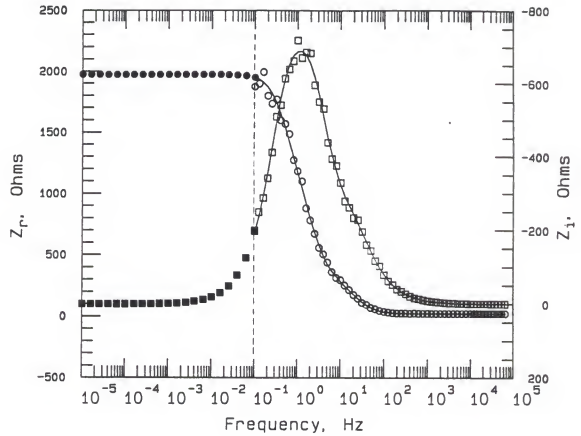


Figure 2.10: Calculated low-frequency impedance for *good-noise*: a data set which exhibited randomly distributed relative residuals for the equivalent circuit fit. This data set is shown to satisfy the KK-relations, with $\omega_0 = 10^{-5}$ Hz and $\omega_{min} = 0.1$ Hz. (*Solid line*) Theoretical impedance response; (○) Noisy real impedance data; (□) Noisy imaginary impedance data; (●) Calculated real component of impedance; and, (■) Calculated imaginary component of impedance. Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

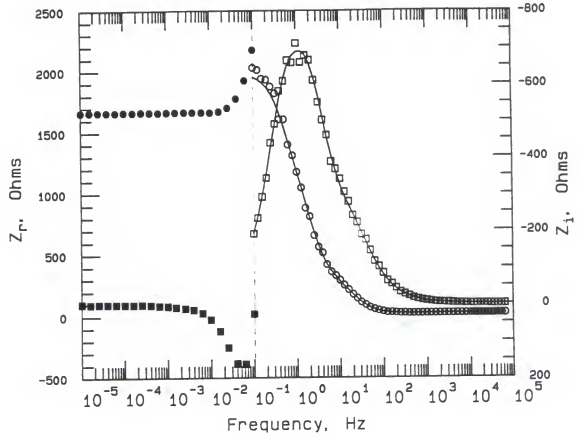


Figure 2.11: Calculated low-frequency impedance for *bad-noise*: a data set which exhibited nonrandomly distributed relative residuals for the equivalent circuit fit. The data set is deemed to be inconsistent with the KK-relations, with $\omega_0 = 10^{-5}$ Hz and $\omega_{min} = 0.1$ Hz. (*Solid line*) Theoretical impedance response; (o) Noisy real impedance data; (□) Noisy imaginary impedance data; (•) Calculated real component of impedance; and, (■) Calculated imaginary component of impedance. Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

2.5.3 Invalid Data Sets

The examples presented in this section are for nonstationary systems. Synthetic data were generated for the equivalent circuit shown in Figure 2.2d by making the value of the resistor R_1 or the capacitor C_1 vary with time (see Appendix A.3.5). The "experimental" frequency range was 0.1 Hz to 65 kHz. Ten data points were collected per decade of frequency, and the number of delay cycles at each frequency N was six. Increasing degrees of inconsistency were used by changing the value of the "inconsistency factor" f. For a variable R_1, the values of f equal to ± 0.1, ± 0.175, ± 0.25, and ± 0.5 correspond to a difference of $\pm 7.0\%$, $\pm 12.2\%$, $\pm 17.4\%$, and $\pm 35.0\%$, respectively, between the initial and final values of R_1 (see equation (A.33), Appendix A). For a variable C_1, the values of f equal to $\pm 10^{-8}$, $\pm 5 \times 10^{-8}$, and $\pm 10^{-7}$ correspond to a difference of $\pm 15.5\%$, $\pm 77.5\%$, and $\pm 155\%$, respectively, between the initial and final values of C_1.

2.5.3.1 Observation of a discontinuity at ω_{min} for inconsistent data

The calculated low frequency impedance values for a system with a variable R_1 are presented in Figures 2.12 and 2.13, while those for a variable C_1 are shown in Figures 2.14 and 2.15. The algorithm has "forced" consistency of the data sets with the KK–equations at the expense of continuity at ω_{min} between the calculated values and the experimental data. The degree of inconsistency is evident from the extent of discontinuity at the truncation frequency. It was observed that the calculations were very sensitive to the value of the lowest frequency ω_0. For a given invalid data set, the discontinuity at ω_{min} became more pronounced as ω_0 was decreased. This is in contrast to the results obtained for consistent data; *i.e.,* the calculated impedance at the low frequency range approached the theoretical impedance as the value of ω_0 was lowered.

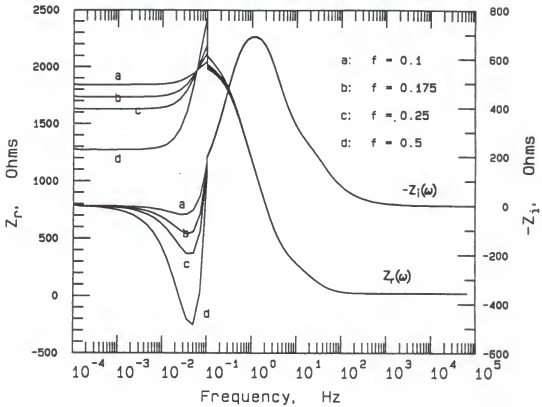


Figure 2.12: Calculated low-frequency impedance for a simulated nonstationary system in the frequency range 10^{-4} Hz to 0.1 Hz. The electrical circuit of Figure 2.2d was used with a variable resistor R_1 and positive values of "inconsistency factor" f. The value of the resistor increased during the course of the "experiment." a: $f = 0.1$ ($\Delta R_1 = 7.0\%$), b: $f = 0.175$ ($\Delta R_1 = 12.2\%$), c: $f = 0.25$ ($\Delta R_1 = 17.4\%$), d: $f = 0.5$ ($\Delta R_1 = 35.0\%$). Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

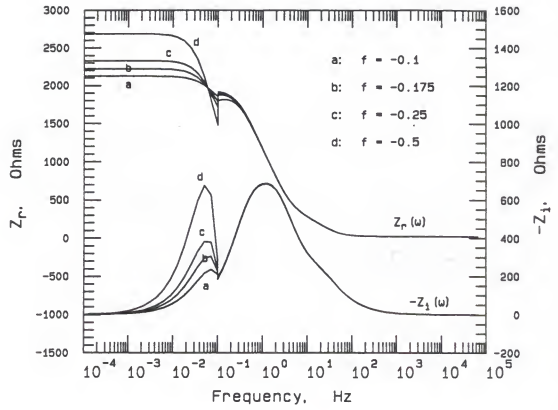


Figure 2.13: Calculated low-frequency impedance for a simulated nonstationary system in the frequency range 10^{-4} Hz to 0.1 Hz. The electrical circuit of Figure 2.2d was used with a variable resistor R_1 and negative values of "inconsistency factors" f. The value of the resistor decreased during the course of the "experiment". **a**: $f = -0.1$ ($\Delta R_1 = -7.0\%$), **b**: $f = -0.175$ ($\Delta R_1 = -12.2\%$), **c**: $f = -0.25$ ($\Delta R_1 = -17.4\%$), **d**: $f = -0.5$ ($\Delta R_1 = -35.0\%$). Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

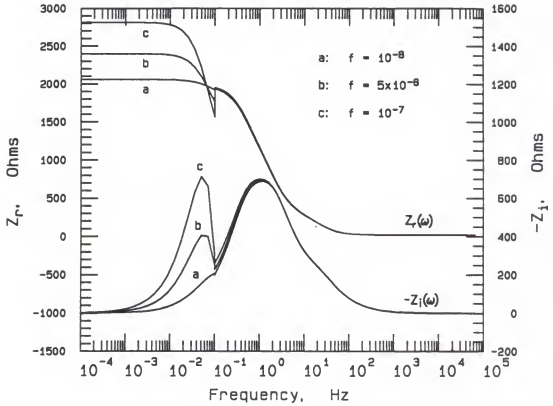


Figure 2.14: Calculated low-frequency impedance for a simulated nonstationary system in the frequency range 10^{-4} Hz to 0.1 Hz. The electrical circuit of Figure 2.2d was used with a variable capacitor C_1 and positive values of "inconsistency factor" f. The value of the capacitor increased during the course of the "experiment". a: $f = 10^{-8}$ ($\Delta C_1 = 15.5\%$), b: $f = 5 \times 10^{-8}$ ($\Delta C_1 = 77.5\%$), c: $f = 10^{-7}$ ($\Delta C_1 = 155\%$). Reprinted by permission of the publisher, *The Electrochemical Society, Inc:* [50].

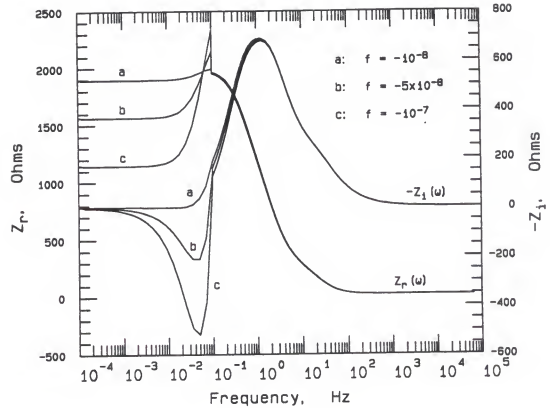


Figure 2.15: Calculated low-frequency impedance for a simulated nonstationary system in the frequency range 10^{-4} Hz to 0.1 Hz. The electrical circuit of Figure 2.2d was used with a variable capacitor C_1 and negative values of "inconsistency factor" f. The value of the capacitor decreased during the course of the "experiment". **a**: $f = -10^{-8}$ ($\Delta C_1 = -15.5\%$), **b**: $f = -5 \times 10^{-8}$ ($\Delta C_1 = -77.5\%$), **c**: $f = -10^{-7}$ ($\Delta C_1 = -155\%$). Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

By employing equivalent electrical circuits to simulate nonstationary systems, the sensitivity of the algorithm to test for the violation of the stability criterion is illustrated in Figures 2.12 to 2.15. Even at the lowest degree of inconsistency, the data sets were deemed to be invalid by the appearance of a distinct discontinuity at ω_{min}. The sensitivity of the proposed KK–algorithm to test for violations of the causality and linearity conditions were not investigated because the impedance data of the electrical circuits listed in Figure 2.2 are inherently linear and causal. Macdonald and Urquidi-Macdonald [48] have stated that their KK–algorithm was insensitive to the violation of the linearity condition for the specific case of iron in sulfuric acid. Application of the proposed KK–algorithm to actual experimental data with large amplitude voltage perturbations can be considered in order to investigate its sensitivity to the linearity constraint.

2.5.3.2 Determination of acceptable degree of discontinuity

It is expected that most experimental systems will show some degree of discontinuity since the algorithm is very sensitive to even slight deviations from stationary behavior. To determine bounds for acceptable amounts of discontinuity, the inconsistent data sets of Figure 2.12 were fitted to the equivalent circuit of Figure 2.2d using LOMFP [16]. The fitted values for the circuit elements of Figure 2.2d are given in Table 2.1. The extent to which these parameters are affected depends on the magnitude of the time constants associated with the circuit elements.

The parameters R_W and T_W changed more as compared to the other circuit elements because the dominant time constant in the low–frequency region is associated with the Warburg impedance. This is the same frequency region where the value of the resistor R_1 changed the most; however, the time constant R_1C_1 was significantly less than T_W at any given frequency. Similar results were obtained for other

Table 2.1: Fitted parameters for the equivalent circuit of Figure 2.2d with a variable resistor R_1 using LOMFP[16]. The consistent data set is given for $f = 0.0$. The nonstationary systems with various degrees of inconsistency are given by $f = 0.1$, 0.175, 0.25, and 0.5.

f	R_s, Ω	R_1, Ω	C_1, μF	R_2, Ω	C_2, μF	R_W,Ω	T_W,s
0.0	23.22	418.3	18.8	695.2	92.6	846.4	0.7055
0.100	23.22	419.1	18.7	699.0	92.4	857.3	0.7141
0.175	23.41	420.2	18.7	701.7	92.2	865.8	0.7204
0.250	23.48	421.0	18.6	704.3	92.0	874.3	0.7265
0.500	23.74	423.3	18.4	712.3	91.5	903.73	0.7459

inconsistent data sets (where either C_1 or R_W were allowed to be functions of time) or when the relative magnitude of the time constants was altered. These results were more evident when the dominant time constant was about two orders of magnitude greater than the other time constants.

Acceptable bounds for levels of discontinuity at the truncation frequency were obtained by comparison to changes in the calculated values of the circuit parameters. The results are presented in Figure 2.16, which illustrates the percent difference between the values of the circuit elements for the stationary and nonstationary systems as a function of the percent difference between the calculated and the experimental values for the *real* impedance at ω_{min}. For these nonstationary systems, a discontinuity of less than 15% represents a deviation of less than $\pm 5\%$ for the circuit elements not associated with the dominant time constant. In other words, regression of data showing a 15% discontinuity can be expected to yield circuit parameters associated with electrochemical reaction rate constants within 5% of the values that would apply at the beginning of the experiment. The Warburg component would show a larger

error. This establishes an error bounds for the circuit parameters of impedance data sets deemed to be inconsistent.

2.5.4 The Effect of Truncating High–Frequency Data

The algorithm incorporated an assumption that the high–frequency impedance data have negligible contributions to the integral when the KK–equations are evaluated at low frequencies. There is a limit to the truncation of high–frequency data, however, for this assumption to be valid.

Low–frequency impedance values were calculated for the equivalent circuit shown in Figure 2.2d with decreasing ω_{max}. The results are presented in Figure 2.17. The Bode plots for $\omega_{max} = 10$ kHz show that the calculated low–frequency impedance is continuous with the available data at 0.1 Hz. However, when $\omega_{max} = 1$ kHz, there is a distinct discontinuity at ω_{min} due to the residual errors at the high–frequency extreme.

The maximum frequency ω_{max} must be sufficiently large that the impedance data at the high frequency range do not truncate in the capacitive region. This requirement can be relaxed by assuming polynomial expressions for the real and imaginary impedance in the range $\omega > \omega_{max}$. Additional summation terms and polynomial coefficients would be introduced into equations (A.11) and (A.12) to account for the new intervals at the high–frequency extreme.

2.6 Application to Experimental Data

A common approach for the interpretation and analysis of EIS data is to fit the data to the analytic impedance response of an equivalent electrical circuit [15–19]. Use of an electrical circuit to model the electrochemical system implies that the

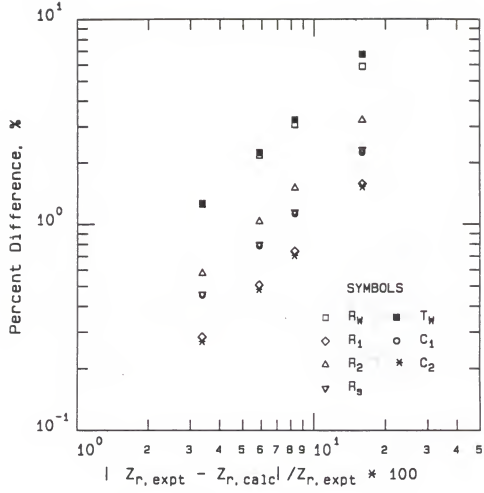


Figure 2.16: The effect on inconsistency on the calculated parameters of the equivalent circuit of Figure 2.2d. The abscissa represents the percent difference between the real component of the data set and the calculated real impedance at ω_{min}. The ordinate is the percent difference between the values of the circuit elements of the nonstationary and stationary systems. Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

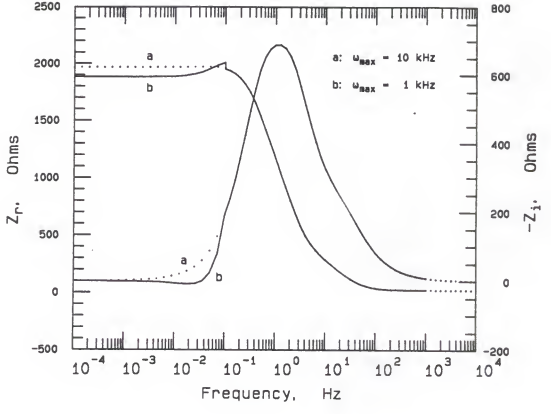


Figure 2.17: The effect of decreasing ω_{max} on the algorithm attributable to the problem of residuals at the high-frequency extreme. The data sets were from the equivalent circuit of Figure 2.2d, with $\omega_0 = 10^{-}4$ Hz and $\omega_{min} = 0.1$ Hz. a: $\omega_{max} = 10$ kHz; and, b: $\omega_{max} = 1$ kHz. Reprinted by permission of the publisher, *The Electrochemical Society, Inc.* [50].

impedance data is characteristic of a system that is linear, causal, and stable. Corroding systems, however, are inherently "nonstationary" due to a continually changing electrode surface and electrochemical environment. Consequently, a corrosionist should be especially interested in determining whether the experimental data obtained for such systems are "valid" for fitting to an equivalent electrical circuit. The validity of EIS data can be inferred from the residuals between the data set and the analytic impedance; a good model fit should exhibit a truly random distribution for the residuals. However, the process of identifying an appropriate electrical circuit can often be tedious and time–consuming, especially when the consistency of impedance data is questionable. The KK–relations offer a straight-forward approach for evaluating the consistency of experimental data without the use of a fitting procedure. It has been shown that equivalent electrical circuits used to model electrochemical systems satisfy the relations [38]. If the consistency of experimental data can be verified using these equations, it follows that an equivalent circuit could be found to model the experimental data with an acceptable "goodness of fit." The treatment of experimental data using a KK–algorithm should complement fitting procedures because the suitability of the data set for fitting to an equivalent circuit can be verified.

In the previous section, the algorithm was shown to be sensitive to failures of the stability criterion of the KK–relations [50]. The application of this algorithm is demonstrated here for a system that is artificially forced to be nonstationary. In addition to determining whether a data set represents a stable system, the algorithm can be used to identify the "cut-off" frequency below which the data show signs of instability. This can be useful when impedance values are obtained sequentially from high to low frequency. Finally, the algorithm is applied to data for a slowly changing system to identify whether the time scale of the experiment is sufficiently short to provide valid data.

2.6.1 Experimental Procedure

Impedance data were obtained from two electrochemical systems: the hydrogenation of a metal hydride electrode in a strongly basic solution and a rotating copper disk electrode at open circuit potential in a chloride solution. A brief description of each system is given below.

- *Hydrogenation of $LaNi_5$.* The impedance data presented for this system are part of a project to determine the applicability of EIS techniques for studying the kinetic and mass transfer processes in metal hydride systems [52]. The electrode was an intermetallic $LaNi_5$ ingot encased in laminating resin epoxy. A silver-soldered nickel wire provided electrical contact. The epoxy was sanded to expose a cross-section of the ingot, and the cross-section was polished using a 600 grit SiC paper and 1 μm alumina powder. A 31 wt% KOH aqueous solution was deaerated with argon for 20 minutes prior to the experiments and intermittently between each frequency scan. Before the impedance measurements were performed, the $LaNi_5$ ingot electrode was cathodically cycled from -0.5 to -1.2V (Hg/HgO) at a sweep rate of 1 mV/sec to form the metal hydride. The impedance measurements were performed at -1.1V (Hg/HgO).

- *Rotating copper-disk electrode.* The impedance data obtained for the copper-disk electrode system are part of a corrosion study for copper and aluminum bronzes in alkaline saline solutions [53]. The electrodes were constructed from high purity copper rods (99.999% Cu) which had been annealed under argon at 500°C for five to eight hours and allowed to cool in the furnace. Each rod was soldered onto a stainless steel shaft and press-fitted into a teflon casing to form the copper-disk electrode with an exposed cross-sectional area of 0.385 cm^2. These electrodes were wet polished with increasingly finer SiC papers followed by 1μm alumina powder to produce a mirror-like surface. The electrodes were

cleaned ultrasonically in deionized water for one minute immediately before the experiment. The pH of the 0.5N chloride aqueous electrolyte was regulated with a Cole–Parmer pH controller Model 5652 using NaOH and HCl solutions of 0.5N chloride concentration. The impedance measurements were performed at open circuit potential.

The EIS instrumentation included a Schlumberger Solartron 1250 frequency response analyzer coupled to either the Solartron 1286 electrochemical or the Princeton Applied Research Model 273 potentiostat. A Pine analytical rotator was employed for the rotating copper-disk experiments.

2.6.2 Analysis of Experimental Data

Corrosion systems are inherently nonstationary because the electrode surface and the electrochemical environment will change during the course of an experiment. The impedance measurements will be influenced by the nonstationary behavior, and these effects are more important at low frequencies where it takes a long time to perform a single measurement. It follows that the impedance response of such systems may fail to meet the stability criterion required by the KK–relations.

The "validity" of experimental impedance data is often determined from the reproducibility of the measurement, either by replicating the results or by scanning back to the starting frequency. Stationary systems will give the same impedance response for any number of measurements performed. One is often asked, however, to interpret unreplicated impedance data. Furthermore, there are cases where one may ask whether a system may be nonstationary over the time required for two frequency scans yet be sufficiently stable over the time required for the first frequency scan.

The application of the KK–algorithm to evaluate the consistency of experimental impedance data obtained for the electrochemical systems described above is presented in this section. The objectives are listed below.

- *The sensitivity of the algorithm to violations of the stability criterion for impedance data of corroding systems will be illustrated.* Using impedance data derived from equivalent circuits, the algorithm has been shown to be a sensitive test for violations of the stability criterion [50]. In this section, nonstationary conditions were simulated by changing the temperature or pH of the electrolyte solution during the course of an experiment using a copper-disk electrode. The appearance of a distinct discontinuity between the calculated impedance and experimental data at the lowest measured frequency ω_{min} is shown to be characteristic of a system which has violated the stability condition of the KK–relations.

- *The algorithm will be used to identify a cut-off frequency for which the experimental impedance data are consistent.* Reproducibility of impedance data might be difficult to achieve for corroding systems such as the metal hydride electrode described above which tends to break apart due to the infusion of hydrogen to the bulk of the electrode at cathodic potentials. Although the frequency scans are not reproducible, a cut-off frequency can be identified for individual scans for which the experimental data are deemed to be consistent. This approach is applied to the experimental data of the metal hydride and copper systems.

- *The numerical difficulties and constraints of the present algorithm will be discussed.* The KK–algorithm presented here involves the adjustable parameters ω_0, K, and M. The parameters K and M are polynomial orders of the approximating functions in the low-frequency domain $\omega_0 \leq \omega < \omega_{min}$ given by

equations (A.3) and (A.4), respectively, in Appendix A. The values of these parameters influence the accuracy of the calculations. An understanding of the limitations of the algorithm facilitates the analysis of experimental impedance data.

2.6.2.1 Nonstationary systems: variable pH and temperature

Impedance data were obtained for a copper disk electrode rotating at 1000 RPM immersed in a 0.5N chloride solution, pH = 9.4. The temperature or pH of the electrolyte solution was allowed to change during the course of an experiment to force a nonstationary condition. The effect of changing electrolyte pH on the value of the open circuit potential for this electrode system is illustrated in Figure 2.18; a change in surface film composition is believed to take place at about a pH of 9.1 [54]. A drift in the open circuit potential will give rise to an overpotential because EIS instrumentation used measured the open circuit potential just before the experiment was begun. The potentiostat was automatically set for this potential, and the ac signal was added to this value [55]. Significant changes in the open circuit potential will result in an overpotential being applied at the interface. The instability of the electrode will influence the consistency of impedance data, especially at low frequencies where is takes a long time to make a measurement.

Impedance measurements were performed at open circuit potential, about −330 mV (*vs.* SCE) measured at the start of the experiment, using a 10 mV perturbation. The temperature and pH profile with their corresponding impedance data are presented in Figures 2.19 and 2.20, respectively. Note that the data are not excessively noisy, and that visual inspection of the data of Figure 2.19 would lead to the conclusion that the system was not stationary, whereas the data of Figure 2.20 are not obviously unstable.

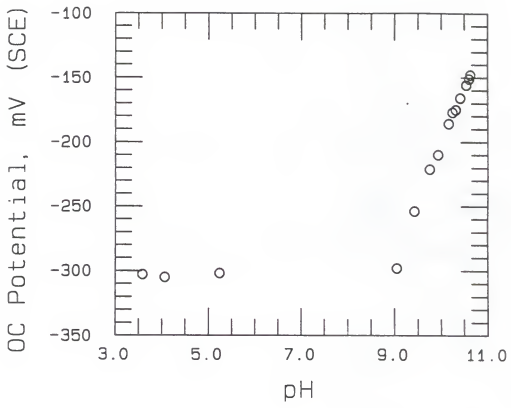


Figure 2.18: Effect of pH on the open circuit potential of a copper–disk electrode rotating at 1000 RPM in a 0.5N chloride solution, 25°C. Reprinted from reference [51].

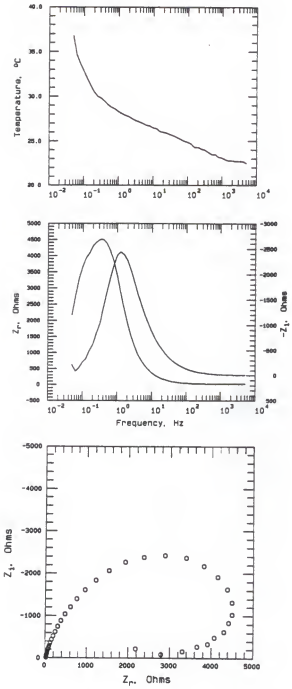


Figure 2.19: Simulated nonstationary system with variable electrolyte temperature. Impedance data obtained for a copper–disk electrode rotating at 1000 RPM in a 0.5N chloride solution, pH = 9.4. The frequency scan, from 5 kHz down to 0.05 Hz, was performed at the open circuit potential, -330 mV (SCE), with a 10 mV perturbation and a 10 second pause between measurements at each frequency. **a)** Measured temperature as a function of frequency; **b)** Bode plot,real and imaginary impedance; and, **c)** Impedance plane plot. Reprinted from reference [51].

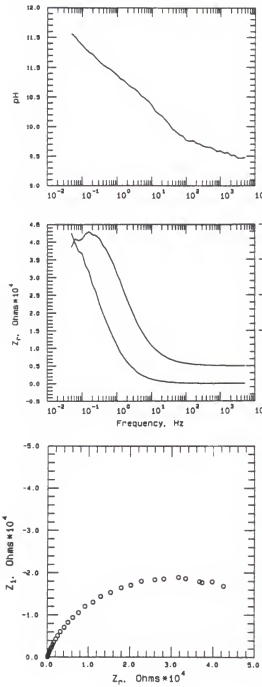


Figure 2.20: Simulated nonstationary system with variable electrolyte pH. Experimental system as in Figure 2.19 except for a constant temperature of 26°C. The initial pH of the solution was 9.4, and the open circuit potential was -215 mV (SCE). **a)** Measured pH as a function of frequency; **b)** Bode plot, real and imaginary impedance; and, **c)** Impedance plane plot. Reprinted from reference [51].

The KK–algorithm was applied to these data sets, and the impedance values below the lowest measured frequencies were calculated as shown in Figures 2.21a and 2.22a for the variable-temperature and the variable-pH experiment, respectively. The calculated real and imaginary impedance below the lowest measured frequency ω_{min} represent the impedance functions that would force the experimental data to satisfy the KK–relations over the "extended" frequency domain $\omega_0 \leq \omega \leq \omega_{max}$. The distinct discontinuity between the calculated impedance values and the experimental data at ω_{min} is indicative of the inconsistency of the experimental data which can be attributed to the violation of the stability condition.

If lower–frequency data were "deleted" (*i.e.*, as if the experiment were prematurely terminated), the "truncated" data set could be tested for consistency and a cut-off frequency identified where the data set could possibly satisfy the KK–relations. This procedure was applied to the data sets of Figures 2.19 and 2.20 up to a frequency where sufficient information about the dominant time constant in the lower-frequency domain was available; this corresponds to the frequency where a maximum in the imaginary impedance is observed. The distinct discontinuity between the calculated impedance and the data sets at ω_{min} is still evident in Figures 2.21b and 2.22b, and these "truncated" data sets are still deemed to be inconsistent. Although the results shown in Figures 2.21c and 2.22c show a discontinuity for the calculated imaginary impedance, an experimenter could erroneously conclude that the "truncated" data sets for these figures are consistent. This issue will be considered in a later section which addresses the numerical difficulties and limitations of the present algorithm.

2.6.2.2 Consistent data: identification of cut–off frequency

Even when impedance measurements are conducted over broad frequency domains that extend to mHz values, low-frequency data are sometimes discarded when their

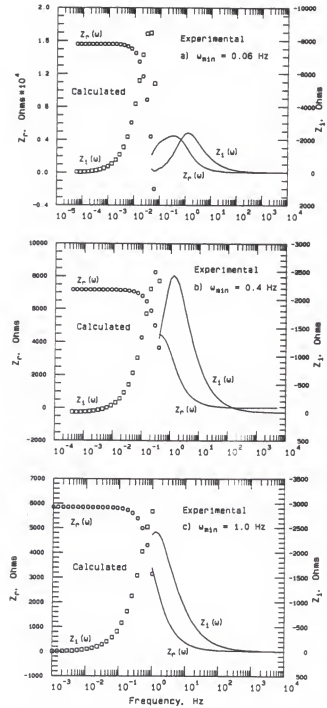


Figure 2.21: Calculated low-frequency impedance for the nonstationary system of Figure 2.19 with variable solution temperature. Figures 2.21a to 2.21c represent the same data set, except that the KK-algorithm was applied to the impedance data at increasing truncation frequencies. The discontinuity between the calculated impedance (symbols ⊙ and □) and the experimental data (solid line) at the truncation frequencies is characteristic of an inconsistent data set. **a)** $\omega_{min} = 0.06$ Hz; **b)** $\omega_{min} = 0.4$ Hz; and, **c)** $\omega_{min} = 1.0$ Hz. Reprinted from reference [51].

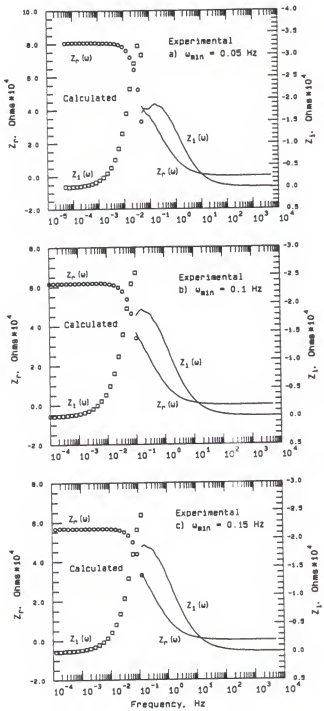


Figure 2.22: Calculated low-frequency impedance for the nonstationary system of Figure 2.20 with variable solution pH. Figures 2.22a to 2.22c represent the same data set, except that the KK-algorithm was applied to the impedance data at increasing truncation frequencies. The discontinuity between the calculated impedance (symbols ⊙ and □) and the experimental data (solid lines) at the truncation frequencies is characteristic of an inconsistent data set. **a)** $\omega_{min} = 0.05$ Hz; **b)** $\omega_{min} = 0.1$ Hz; and, **c)** $\omega_{min} = 0.15$ Hz. Reprinted from reference [51].

validity is questioned (see, for example, Silverman [56]). The decision to discard data, however, requires an understanding of the processes occurring at the electrode surface. The identification of a cut-off frequency, for which the experimental data within a finite frequency domain is known to satisfy the KK–relations, provides a rationale for excluding data from an equivalent circuit fit. The application of the KK–algorithm for this purpose is illustrated in this section.

The experimental data used here were obtained for the hydrogenation of a metal hydride electrode in 31 wt% KOH solution at -1.10 V (Hg/HgO) and for a rotating copper-disk electrode in a saline solution at the open circuit potential. The temperature and pH of the electrolyte solutions used in these experiments were carefully maintained using a water bath and an automatic pH-controller. Several frequency scans, ranging from 65 kHz down to 0.1 Hz, were performed consecutively, and three data sets for each system were chosen for analysis using the proposed KK-algorithm.

Impedance plane plots for the data sets of the metal hydride system are shown in Figure 2.23. The solid lines represent three consecutive frequency scans which are labeled A, B and C. The irreproducibility of the data sets reflects the nonstationary behavior of the electrochemical system due possibly to the changes in diffusivities induced by the infusion of hydrogen into the electrode. Application of the KK–algorithm to these impedance data, which extend to a frequency of 0.1 Hz, indicate that all the experimental data sets are inconsistent as shown in Figure 2.24. When lower frequency data were discarded, a cut-off frequency was identified for each data set. The impedance data above this cut-off frequency satisfy the KK–relations, and these results are illustrated for Data sets A–C in Figure 2.25. The calculated real and imaginary impedance below these cut-off frequencies are shown in the impedance plane plot of Figure 2.23 as symbols $\odot$, $\square$ and $\diamond$ for Data sets A, B and C, respectively. These cut-off frequencies represent the departure from a "stationary"

(or nearly stationary) behavior, *i.e.,* where the time-scale of the measurement at a given frequency is sufficiently long to influence the processes occurring at the electrode surface. Identification of such a cut-off frequency for a given frequency scan provides a valid "snapshot" for time-dependent processes.

Impedance plane plots for the data sets of the copper-disk electrode system are shown in Figure 2.26. The solid lines represent three consecutive frequency scans which are labeled I, II and III. The nonstationary behavior of the electrochemical system is manifested through the irreproducible frequency scans and has been attributed to the growth of a film on the electrode surface [53]. In addition, the change in the chemical nature of the electrode interface results in a drift of the open circuit potential. Since EIS instrumentation measures this potential at the start of an experiment and applies it throughout the spectrum [55], a nonstationary open circuit potential will give rise to an overpotential. The magnitude of this overpotential is given by the difference between the initial potential and the true open circuit potential at some time during the experiment. The KK–algorithm was applied to these impedance data which extend to a frequency of 0.1 Hz. The results suggest that Data sets I and II are inconsistent while Data set III satisfies the KK–relations as shown in Figure 2.27. When the lower frequency data are discarded, a cut-off frequency can be identified for Data sets I and II which delimits the frequency domain of the experimental data as presented in Figure 2.28. For Data set III, which has been shown to be consistent, the algorithm yielded comparable results for the calculated impedance when the entire data set was used and when low-frequency data were discarded. The calculated real and imaginary impedance below these cut-off frequencies are shown in the impedance plane plot of Figure 2.26 as symbols $\odot$, $\square$ and $\diamond$ for Data sets I, II and III, respectively.

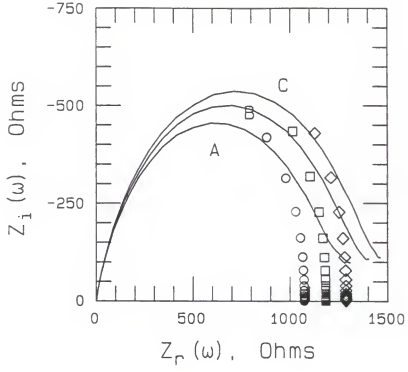


Figure 2.23: Impedance plane plot for the hydrogenation of a metal hydride electrode in 31 wt% KOH solution, −1.10 V (Hg/HgO). (Note the difference in scales for the axes). Curves A, B, and C represent successive frequency scans in the range of 65 kHz down to 0.1 Hz. The symbols ⊙, □, and ◇ are the corresponding low-frequency impedance calculated from the KK-algorithm as shown in Figure 2.25. Reprinted from reference [51].

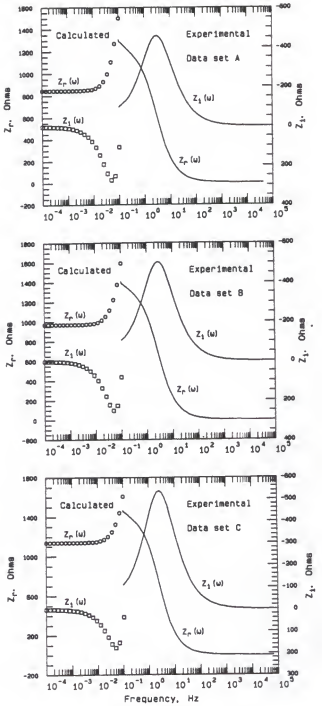


Figure 2.24: Calculated low-frequency impedance values for Data sets A, B, and C of the metal hydride system. These figures illustrate the results of the KK–algorithm as applied to the entire data set and shows the inconsistency of the available data due to the instability of the electrode. of the system. Reprinted from reference [51].

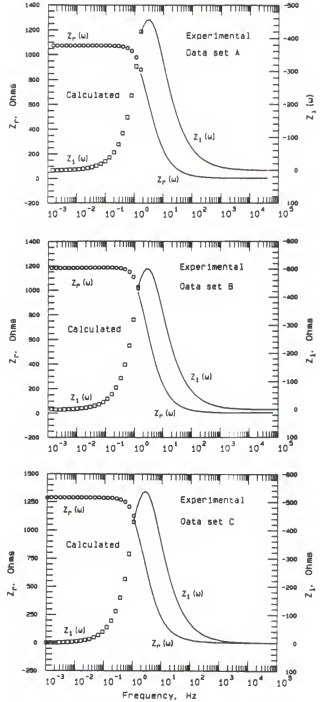


Figure 2.25: Calculated low-frequency impedance values for Data sets A, B, and C of the metal hydride system. These figures show the results of the KK-algorithm after the lower-frequency data are "deleted" to make the remaining data satisfy the KK-relations. Although the successive frequency scans are shown to be characteristic of a nonstationary system in Figure 2.24, these figures show that part of each frequency scan represent a valid "snapshot" of the system. Reprinted from reference [51].

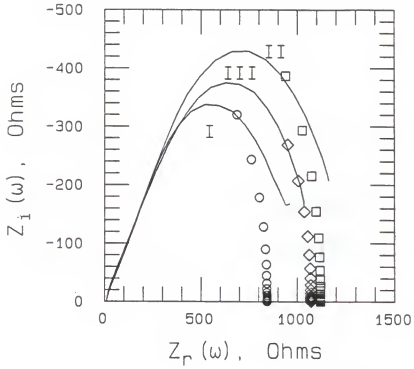


Figure 2.26: Impedance plane plot for a rotating copper-disk electrode at 1000 RPM in a 0.5N chloride solution at the open circuit potential, pH = 9.4. Curves I, II, and III represent successive frequency scans in the range of 65 kHz down to 0.1 Hz. The symbols ⊙, □, and ◇ are the corresponding low-frequency impedance calculated from the KK-algorithm as shown in Figure 2.28. Reprinted from reference [51].

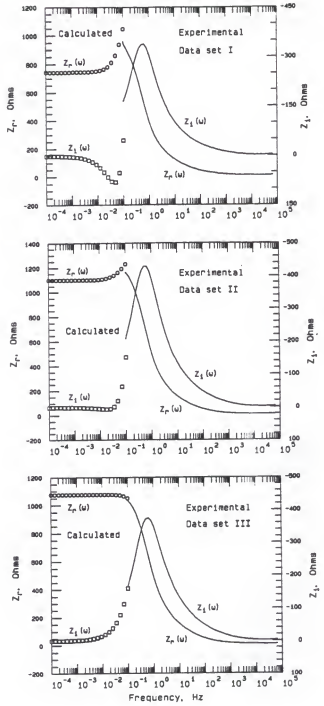


Figure 2.27: Calculated low-frequency impedance values for Data sets I, II, and III of the rotating copper-disk system. As in Figure 2.24, the results of the KK-algorithm when applied to the entire data set show the inconsistency of the available data. Reprinted from reference [51].

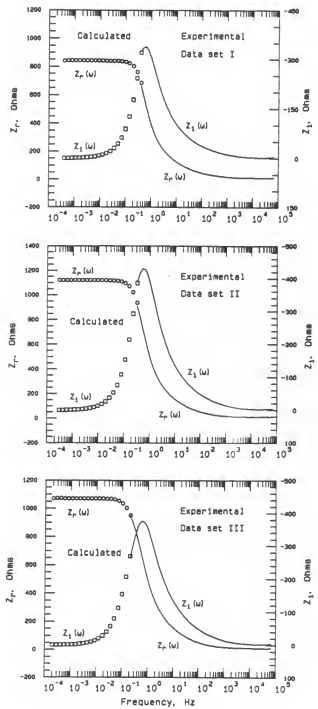


Figure 2.28: Calculated low-frequency impedance values for Data sets I, II, and III of the rotating copper-disk system after deletion of several low-frequency data points. As in Figure 2.25, a truncation frequency can be identified for each data set above which the impedance data is known to satisfy the KK-relations. Reprinted from reference [51].

2.6.3 Implications for Equivalent Circuit Fitting

An important parameter for a corrosionist is the low-frequency asymptote for the real impedance which is required to estimate a polarization resistance. For experimental data that do not extend to a DC limit, an experimenter usually extrapolates the data using the fitting parameters of an equivalent circuit which reasonably models the electrochemical system. The identification of an appropriate equivalent circuit, however, often makes such fitting algorithms tedious and time–consuming, especially when the consistency of the data set is questionable. Moreover, the suitability of experimental data for treatment by a fitting procedure to an equivalent circuit is not ascertained until after the residuals between the experimental and theoretical impedance have been computed and analyzed; a "good" model fit should exhibit truly random residuals.

The advantage of the KK–algorithm presented here is that an equivalent circuit fit is not required to determine the consistency of experimental data. A cut-off frequency for the experimental data set can be identified above which the impedance data are shown to satisfy the KK–relations. Consequently, impedance data above this cut-off frequency should be used for the equivalent electrical circuit fit. In addition, the calculated impedance values below the cut-off frequency provide a conservative estimate of the DC limit for data which are deemed to be consistent using the criterion of this KK–algorithm. The KK–algorithm has been shown to calculate low-frequency impedance within 5% of the theoretical values when the experimental data provides enough information about the dominant time constant in the lower-frequency domain [50]. This is illustrated here for Data sets A–C of the metal hydride system which were fitted to the impedance response of the equivalent electrical circuit shown in Figure 2.29 using LOMFP [16]. Experimental data above the cut-off frequencies identified in Figure 2.25 were used for the fitting procedure, and the values of the

Table 2.2: Fitting parameters for the equivalent circuit of Figure 2.29 using Data sets A–C of the metal hydride system. The fitting procedure used was LOMFP [16].

Data Set	R_s, Ω	R_1, Ω	C_1, μF	R_2, Ω	C_2, μF	R_W,Ω	T_W,s
A	3.812	248.3	19.46	523.1	25.20	351.6	0.133
B	3.825	269.9	19.64	570.1	25.26	393.6	0.144
C	4.117	309.0	20.52	629.5	27.18	384.8	0.170

Table 2.3: Comparison of the low-frequency asymptotes for the real impedance obtained from the KK–algorithm and the extrapolation of theoretical impedance using the fitting parameters for the circuit shown in Figure 2.29

Data Set	Equivalent Circuit	KK–algorithm	Difference
	Ω	Ω	%
A	1127	1073	4.8
B	1237	1184	4.3
C	1327	1288	2.9

fitting parameters are presented in Table 2.2. The theoretical impedance response obtained from these fitting parameters are shown as solid lines in Figure 2.30. These figures show good fit between the experimental data ($\odot$ and $\square$) and the theoretical values (solid line) within the experimental frequency domain. Furthermore, the calculated impedance values for the KK-algorithm ($\odot$ and $\square$) are comparable with the extrapolated values of the equivalent circuit fit, that is, within 5% of the theoretical values as presented Table 2.3. It follows that the low–frequency asymptote calculated by the KK–algorithm for Data sets I–III should provide reasonable estimates of the polarization resistances for the copper-disk electrode system, as illustrated in the impedance plane plot of Figure 2.26.

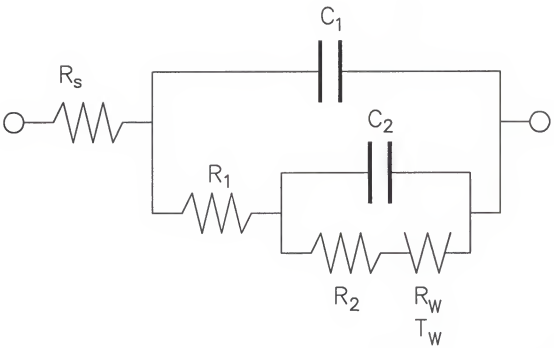


Figure 2.29: Equivalent electrical circuit used for the fitting procedure LOMFP [16] for Data sets A–C of the metal hydride system. Reprinted from reference [51].

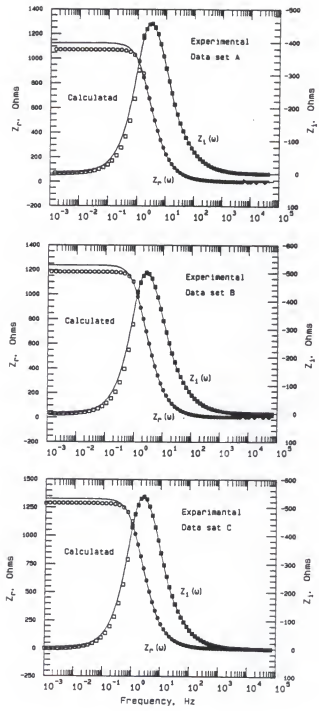


Figure 2.30: Comparison between the calculated impedance values using the KK–algorithm (⊙ and □) and the extrapolated impedance using the fitting parameters of an equivalent circuit fit (solid line), where the symbols ● and ■ represent the experimental Data sets A–C of the metal hydride system. Reprinted from reference [51].

2.6.4 Numerical Constraints

The algorithm presented here involves several adjustable parameters, and the proper choice for their values will influence the accuracy of the calculations. These adjustable parameters are the lower frequency limit ω_0 and the orders of the approximating functions K and M. (Other adjustable parameters are associated with the numerical integration using a Simpson's rule algorithm given in Appendix A [50]. The number of integration segments and segment divisions were sufficiently large to minimize round-off and integration errors and are not discussed further here.) The numerical constraints associated with the KK–algorithm are summarized below.

- *Limits to the value of* ω_0. Although it is numerically feasible to extrapolate to lower frequencies, there is a reason for limiting the value of ω_0. A conservative extrapolation is prudent because the contributions to the integrals of KK–relations originate near the frequency being evaluated [25, 29], that is, within three or four orders of magnitude of that frequency. Since the object of the algorithm is to test the consistency of the experimental data, the calculated impedance should extend below ω_{min} only as far as the available data could contribute to the integrals of the KK–relations. Therefore, the value of ω_0 was limited to three orders of magnitude below ω_{min} for all the calculations presented in this paper.

- *Values of the polynomial orders* K *and* M. The order of the approximating functions K and M must be sufficient to establish the behavior of the impedance over several decades of frequency. Increasing the value of the polynomial orders increases the number of points to be calculated within the low-frequency region, which is given by $K + M + 2$. For synthetic data obtained from equivalent electrical circuits, it has been observed that larger values of K and M yield more

accurate estimates for the calculated impedance. Values for K and M between 6 and 10 were found to be adequate, and 9 was chosen for all calculations.

- *Deletion of experimental impedance data at low frequencies.* The utility of the KK–algorithm to identify a cut-off frequency for each data set was shown in Figures 2.23 to 2.28. However, indiscriminate deletion of available data is discouraged because the accuracy of the calculations is affected as the available data move deeper into the capacitive region since information about the dominant time constant in the low-frequency domain is lost. The results of the algorithm are most reliable when the imaginary impedance exhibits a maximum within the frequency domain being analyzed.

- *Insensitivity to inconsistency at high frequency.* The KK–algorithm, as presented in Appendix A, is more sensitive for testing inconsistencies at the lower frequencies than at the higher frequencies. It can be seen from equations (2.6) and (2.7) that the contributions of the higher frequency data to the integrals diminish almost as rapidly as $1/x^2$, where x is the integration variable. Consequently, the equations presented in Appendix A are relatively insensitive to inconsistencies in the high frequency domain. A possible extension of this work is to calculate the impedance values above the maximum measured frequency ω_{max}. This may provide a more sensitive test for consistency of the high frequency data.

CHAPTER 3
PHOTODISSOLUTION OF GaAs

The relative ease of fabricating a semiconductor photoelectrochemical cell is due to the instantaneous formation of a junction. A regenerative solar cell is illustrated schematically in Figure 3.1 for an n-type semiconductor photoanode immersed in an electrolyte solution containing a redox couple A/A^+. The difference between the potential at the surface of the semiconductor and the potential at the current collector, illustrated by the bending of the energy bands, acts to separate the electron-hole pairs generated by illumination. The electrons migrate towards the current collector while holes accumulate at the surface of the solid. The potential difference increases when the semiconductor photoanode is polarized anodically. Under illumination, however, the photogenerated holes which accumulate at the surface will participate in the anodic decomposition of the semiconductor. A redox couple is chosen which can compete effectively for holes such that the anodic reaction at the interface will be the oxidation of the reducing agent instead of the oxidation of the semiconductor itself. The photogenerated electrons are involved in the reduction reaction at the counterelectrode.

3.1 The Semiconductor–Electrolyte Interface

The current-potential characteristics of a semiconductor photoanode is influenced dramatically by processes which can occur at the semiconductor-electrolyte interface

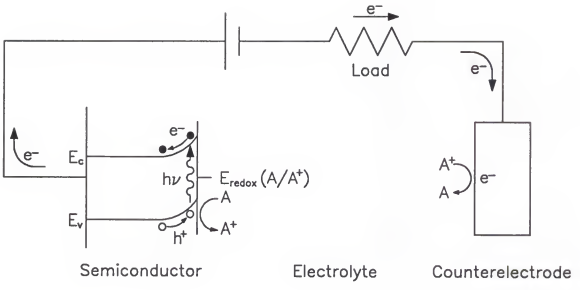


Figure 3.1: Schematic illustration of a regenerative solar cell for an illuminated n-type semiconductor photoanode in an electrolyte solution containing a redox couple A/A^+.

[57–63]. These would include charge redistribution, band-bending, surface recombination, photogeneration of carriers, and charge-transfer reactions. An understanding of how these mechanisms interact is essential to the study of decomposition processes and its stabilization.

3.1.1 Potential Distribution and Photoeffects

When a semiconductor material is immersed in an electrolyte solution, a redistribution of charge results in a potential drop near the interface and the formation of a space-charge layer due to the relatively low concentration of electrons in semiconductors. A schematic illustration of the potential distribution for the semiconductor-electrolyte interface is shown in Figure 3.2. The difference between the potential at the interface and the potential at the bulk of the semiconductor (or current collector) is defined to be the flat-band potential. This potential variation corresponds to a bending of the energy band edges in the space-charge region where, at equilibrium and in dark condition, the Fermi level of the semiconductor E_F and the redox potential of the electrolyte are equal [63]. The influence of external polarization is primarily in the space-charge region resulting in a change in the potential drop across the semiconductor [57, 62]. For n-type materials, anodic polarization increases band bending, while cathodic polarization decreases the band bending; the reverse is observed for p-type materials. The extent of polarization required to "unbend" the energy bands is called the flat-band potential. The potential is uniform throughout the semiconductor at the flat-band potential, including the space charge region. The value of the flat-band potential can be determined from capacity measurements [61, 62, 64] through the Mott-Schottky equation for doped semiconductors

$$\left(\frac{1}{C_{sc}}\right)^2 = \frac{2}{e_o \varepsilon \varepsilon_o [N_d - N_a]} \cdot \left(\Delta\phi_s - \frac{k_B T}{e_o}\right) \tag{3.1}$$

where C_{sc} is the capacity, $[N_d - N_a]$ is the net dopant concentration, $\Delta\phi_s$ is the potential difference, T is the temperature, k_B is the Boltzmann constant, e_o is the elementary charge, ε is the dielectric constant of the semiconductor, and ε_o is the permittivity in vacuum. In addition, the position of the band-edges can also be established from these measurements for a given electrolyte composition and pH [62, 64].

Illumination of the semiconductor-electrolyte interface results in the generation of electron-hole pairs for light energies greater than the bandgap of the semiconductor [61–63]. The photogenerated electrons and holes are separated by the potential gradient (or electric field) in the space charge region. For n-type semiconductors, the photogenerated holes move towards the surface while the electrons move towards the bulk of the semiconductor under the influence of the electric field. This process is enhanced by anodic polarization which increases the potential drop in the space-charge region and, by separating electrons and holes in the space charge region, reduces the rate of electron-hole recombination. For p-type materials, the photogenerated electrons move towards the surface while the holes move towards the bulk of the semiconductor. Cathodic polarization increases the potential drop in the space-charge region and reduces the rate of recombination. The energy band diagram for the illuminated condition is shown in Figure 3.3. The accumulation of minority carriers at the surface under illumination (or at high polarization in the dark) is the primary cause of charge-transfer reactions across the semiconductor-electrolyte interface, that is, with redox species in solution and/or the decomposition of the semiconductor [61]. The anodic dissolution of semiconductors has been identified as a valence band process involving holes at the surface [62]. Gerischer [65] proposed that the overall decomposition reaction of GaAs in a basic medium is given by

$$GaAs + 6p^+ + 10OH^- \longrightarrow GaO_2^- + AsO_3^{-3} + 5H_2O, \qquad (3.2)$$

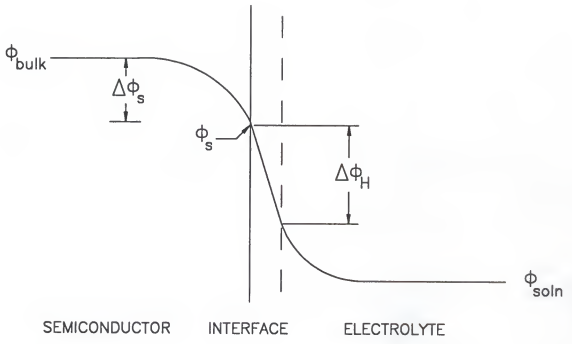


Figure 3.2: Schematic illustration of the potential distribution for a semiconductor-electrolyte interface.

and in an acid medium as

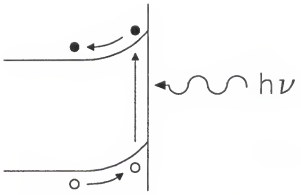
$$GaAs + 6p^{+} + 6X^{-} \longrightarrow Ga^{+3} \cdot aq + AsX_6^{-3} \tag{3.3}$$

or

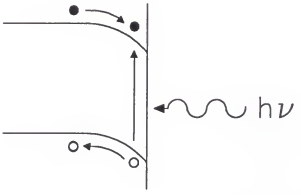
$$GaAs + 6p^{+} + 3H_2O \longrightarrow Ga^{+3} \cdot aq + H_3AsO_3 + 3H^{+} \cdot aq. \tag{3.4}$$

It is also possible to enhance photodecomposition using oxidizing agents, such as $Fe(CN)_6^{3-}$ or Ce^{+4}, through a hole-injection process at the valence band [61,66–69].

Charge-transfer reactions at the interface for the illuminated case give rise to a photocurrent. This is illustrated by the experimental results presented in Figure 3.4, where the steady-state photocurrent for an n-type GaAs disk electrode is given as a function of electrode potential. At potentials more anodic than -0.25V (SCE), the surface concentration of electrons may be assumed to be negligible due to increased band bending and that the rate of recombination is negligible [70]. At lower polarization, recombination at the surface and within the space-charge region become significant due to a decrease in band bending. The influence of recombination is evident in the potential region below -0.25V (SCE) where the photocurrent decreases drastically. Vanmaekelbergh *et al.* [70] stated that, in the case of semiconductor/electrolyte Schottky-barrier electrodes, recombination within the space-charge and neutral region is the predominant process due to rate-limited charge transfer at the semiconductor-electrolyte interface. The photocurrent in this example is due to the anodic decomposition of the GaAs by hole capture since the concentration of the reducing agent Fe^{2+} (5 mM) is too small to compete with the dissolution reaction. The saturation photocurrent is a linear function of light intensity [61, 62]. Similar results are shown in Figure 3.5 for the GaAs-system described above.



a) n–TYPE SEMICONDUCTOR



b) p–TYPE SEMICONDUCTOR

Figure 3.3: Energy band diagram for an illuminated semiconductor-electrolyte interface. The photogenerated electron-hole pairs are separated by the electric field in the space-charge region. a) n-type semiconductor, b) p-type semiconductor.

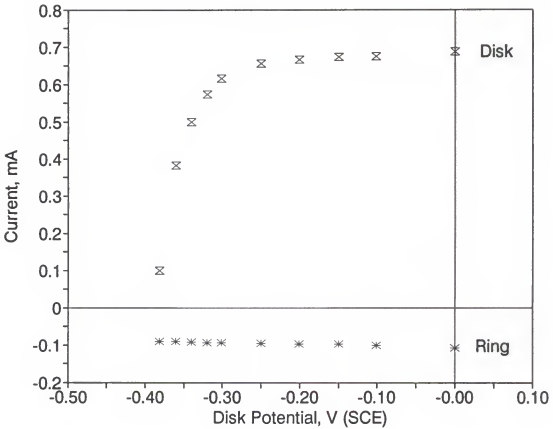


Figure 3.4: Steady-state photocurrent as a function of disk electrode potential for an n-type Si-doped GaAs disk electrode, 0.5cm diameter, rotating at 1000 RPM. The electrolyte solution was 0.25M Na_2SO_4, 5 mM Fe^{2+}, pH = 1.4. The relative light intensity was 91.4% of the maximum intensity shown in Figure 3.5.

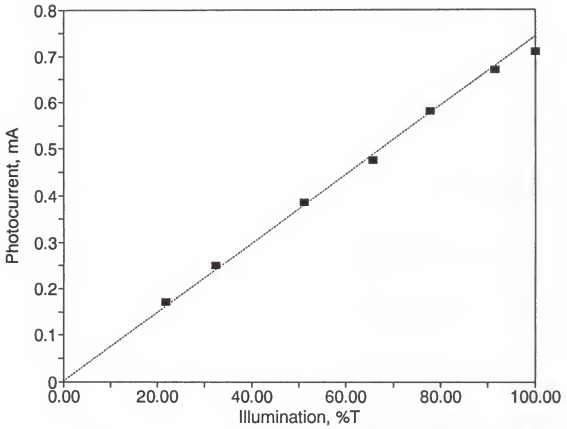


Figure 3.5: Saturation photocurrent as a function of light intensity (percent transmission relative to the maximum intensity). The experimental system is described in Figure 3.4.

3.1.2 Thermodynamic and Kinetic Aspects

The analysis of thermodynamic stability against photocorrosion is based on consideration of the reactions that are possible at the semiconductor-electrolyte interface. It should be noted that while thermodynamics can be used to predict the susceptibility of a photoanode to corrosion, the decomposition processes are still controlled by the charge transfer kinetics across the interface, influenced by the solution composition and by the properties of the semiconductor solid. Suitable redox couples have been shown to compete effectively with photodecomposition. The thermodynamic aspect of semiconductor dissolution for the ideal, or defect-free, case has been discussed by Bard and Wrighton [71] and Gerischer [72, 73]. The thermodynamic and kinetic model of Frese *et al.* [74, 75] suggests that surface defects act as initiation points for the corrosion process due to broken bonds resulting in a more negative "microscopic" decomposition potential relative to the decomposition potential for the ideal case. A similar model was also described by Gerischer [73]. Park and Barber [76] have presented Pourbaix (or predominance) diagrams for several semiconductors. A summary of their discussion is given in this section.

The corrosion of semiconductors can occur by oxidation involving holes from the valence band or by reduction involving electrons from the conduction band. Upon assuming reaction schemes for these processes, the standard redox potential of the decomposition reactions ϕ_{decomp}, relative to a reference electrode, can be estimated theoretically from the free energy of the reaction paths [73, 74]. The Fermi energy of the redox potential are related by a constant given by

$$E = E_{\text{ref}} - e_o \phi_{\text{decomp}} \tag{3.5}$$

where $E_{\text{ref}} = -4.5$ eV for the normal hydrogen electrode (NHE) [62, 72]. The Fermi energies ${}_nE_{\text{decomp}}$ and ${}_pE_{\text{decomp}}$ correspond to the anodic and cathodic decomposition potentials, respectively. The position of ${}_nE_{\text{decomp}}$ and ${}_pE_{\text{decomp}}$ with respect to the

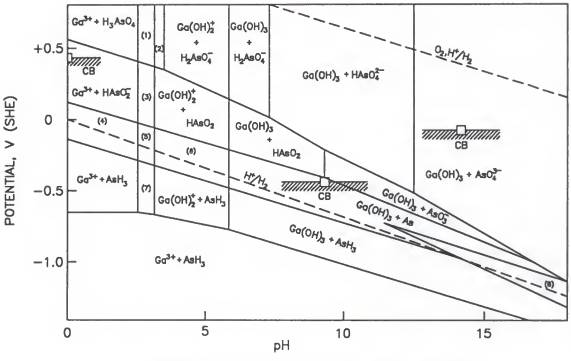
conduction band and valence band edges can be determined, and one can infer whether the semiconductor is susceptible to corrosion. The solid is stable against anodic decomposition when ${}_pE_{\text{decomp}}$ is below the valence band and stable against cathodic decomposition when ${}_nE_{\text{decomp}}$ is above the conduction band [61, 72]. However, for most semiconductor/electrolyte systems, either ${}_nE_{\text{decomp}}$ or ${}_pE_{\text{decomp}}$, or both, lie within the mid-bandgap region, and the solid is thermodynamically susceptible to corrosion. Bard and Wrighton [71] explained the need to consider the position of redox potential relative to the decomposition potential. Hole capture at the surface by the reducing agent is more favored thermodynamically when the redox potential is more negative than the decomposition potential. The Pourbaix diagram for GaAs derived by Park and Barber [76] is illustrated in Figure 3.6. The diagram shows the predominant reaction species at given potentials and solution pH, and the position of the conduction band edges at flat band condition at different pH values are represented by the symbol □. Their calculations predicted a small window of stability at very high pH values and negative potentials, as shown in region 8 in Figure 3.6, and the expected susceptibility of GaAs to anodic decomposition.

Gerischer's criterion for decomposition by charged carriers was based on the relative values of the quasi-Fermi levels for electrons and holes under illumination versus the Fermi energy of the decomposition reactions [61, 72]. At equilibrium and in the dark, the Fermi levels of holes and electrons are equal, and the equilibrium carrier concentrations n_o and p_o are specified by the value of E_F. With illumination, generation of minority carriers results in a significant change in the quasi-Fermi energy level of the minority carrier within the space charge layer as given by

$$ {}_nE_F^* = E_c + k_BT \ln\left(\frac{n_o + \Delta n}{N_c}\right) \tag{3.6} $$

$$ {}_pE_F^* = E_v + k_BT \ln\left(\frac{p_o + \Delta p}{N_v}\right) \tag{3.7} $$

where ${}_nE_F^*$ and ${}_pE_F^*$ are the quasi-Fermi energy of electrons and holes, respectively,



Region	Species Predominant	Region	Species Predominant
(1)	$Ga(OH)^{2+} + H_3AsO_4$	(5)	$Ga(OH)^{2+} + As$
(2)	$Ga(OH)_2^+ + H_3AsO_4$	(6)	$Ga(OH) + As$
(3)	$Ga(OH)^{2+} + HAsO_2$	(7)	$Ga(OH)^{2+} + AsH_3$
(4)	$Ga^{3+} + As$	(8)	GaAs

Figure 3.6: Pourbaix or predominance diagram for GaAs in water by Park and Barber [76]. The □ symbols represent flat band potentials. Reprinted by permission of the publisher, *Journal of Electroanalytical Chemistry and Interfacial Electrochemistry.*

Δn and Δp are the concentration of the photogenerated carriers, and N_c and N_v are the effective density of states in the conduction and valence band, respectively. When the Fermi energy of the minority exceeds the Fermi energy of the decomposition reaction, corrosion is possible [61] by anodic decomposition by holes

$$ {}_pE_F^* \; < \; {}_pE_{\text{decomp}} \tag{3.8} $$

or by cathodic decomposition by electrons

$$ {}_nE_F^* \; > \; {}_nE_{\text{decomp}} \, . \tag{3.9} $$

In the presence of an effective redox couple, the consumption of minority carriers due to charge transfer across the semiconductor-electrolyte interface decreases the quasi-Fermi energy near the surface as compared to the case without redox reactions. A decrease in the concentration of holes at the surface enhances the stability of the semiconductor electrode; therefore, kinetic limitations to redox reactions can play a major role in determining the stability of a semiconductor electrode. Mathematical models of the semiconductor-electrolyte interface have been developed that solve explicitly for the concentration of holes and electrons within the solid and account for the charge transfer reactions across the interface [77–80].

Frese *et al.* [74] suggested that surface defects act as initiation points for corrosion. The disbonding of surface atoms by hole capture results in a more negative decomposition potential at defect sites as compared to the over-all decomposition potential and the redox potential. Their examples illustrated that surface-damaged electrodes exhibited lower stabilization as compared to an etched, "damage-free" surface [74]. Dislocations also serve as carrier traps resulting in high recombination rates and current loss [30]. The current-voltage characteristics and efficiency of a photoanode are significantly improved by etching of the damaged surface [30, 81]. Since the corrosion reaction at defect sites is thermodynamically favored, the redox reaction of

the stabilizing agent must be kinetically favored in order to compete effectively. The decomposition mechanism proposed for compound semiconductors by Gerischer [73], for example, involve multiple bond-breaking steps initiated at kink sites. The corrosion process can be inhibited by the fast kinetics of a redox couple before further irreversible bond-breaking can occur [82]. The work of Vanmaekelbergh *et al.* [83–85] suggest that the stabilization reaction need not be a direct hole-capture from the valence band. In other words, decomposition intermediates can participate to some extent in the stabilization process.

The application of the semiconductor-electrolyte junction for regenerative solar cells requires the mitigation of the decomposition reactions and the minimization of recombination centers. These have been achieved by employing efficient reducing agents such as the Se^{2-}/Se_2^{2-} redox couple [86] and by modifying the semiconductor surface. Effective surface treatment of the semiconductor surface involves metallization by chemisorbed layers of lead(III) [87], ruthenium(III) [87, 88], or silver [89] which reduces the rate of electron-hole recombination and, in some cases, act as electrocatalysts for the desired stabilization reactions. Thin-metal layers such as gold, platinum, or rhobium [90] have been used to isolate the semiconductor from the corrosive action of the electrolyte solution.

3.1.3 Stabilization of Photocorrosion

The rotating ring-disk electrode (RRDE) is the most common tool employed for studying the stabilization of photodecomposition by redox couples. A more detailed description of this apparatus is given in the following section. The semiconductor specimen is fabricated into a disk electrode and arranged in a concentric configuration with a noble metal ring electrode (such as platinum or gold). An insulating gap, made of epoxy or Teflon, separates the disk and ring assemblies. When a reducing

agent in solution competes effectively with the anodic decomposition of the semiconductor material, it is oxidized at the disk electrode by the consumption of holes. The formation of the oxidized species is detected at the ring electrode being held under limiting current condition for the cathodic reaction. The anodic current at the disk electrode corresponding to the competing stabilization reaction is related to the ring current by the collection efficiency of the RRDE. The collection efficiency N_o is a function only of the electrode geometry. The stabilization ratio s is defined as the fraction of the photocurrent density at the disk electrode which corresponds to the competing oxidation of the reducing agent.

The RRDE has been used to study the photodecomposition of GaP and GaAs in electrolytes containing stabilizing agents [70,74,85,91–93]. Possible reaction mechanisms were proposed to explain the observed kinetic laws which relate the stabilization s to the concentration of reducing agent y, the concentration of conduction band electrons at the surface n_s, and the saturation current density i_h due to the diffusion of holes to the surface. The experimental procedures were designed to identify the role of surface recombination. Two types of experiments were employed. First, the anodic potential of the GaAs electrode was held constant at a value where a saturated photocurrent was observed and the extent of stabilization was measured at various light intensities. Under this condition, the effect of recombination is negligible since $n_s \sim 0$. The second type of experiment is performed at constant light intensity while varying the electrode potential in the potential region below the saturation photocurrent. Here, surface recombination becomes a possible reaction step. A summary of the of the work of Vanmaekelbergh *et al.* [85] for an illuminated n-type GaAs electrode using Fe^{2+} as the stabilizing agent is given below.

Experimental results at varying light intensities showed that stabilization s followed the kinetic law

$$\frac{s^2}{1-s} \sim \frac{y^2}{i} \tag{3.10}$$

when surface recombination was negligible, that is, at very anodic potentials [85]. The limiting value of stabilization s was zero at high values of photocurrent i (or illumination intensities). Previous analysis [83, 84] had shown that these observations can be explained by a mechanism where the stabilization reaction occurs through a decomposition intermediate X_1, and this same intermediate is the oxidizing agent for subsequent decomposition steps. The decomposition reactions are given by

$$(\mathrm{GaAs})_{\mathrm{surf}} + h^+ \xrightarrow{k_1^0} X_1 \tag{3.11}$$

$$X_1 + X_1 \xrightarrow{k_2^0} X_2 + (\mathrm{GaAs})_{\mathrm{surf}} \tag{3.12}$$

$$X_2 + X_1 \xrightarrow{k_3^0} X_3 + (\mathrm{GaAs})_{\mathrm{surf}} \tag{3.13}$$

$$\vdots \quad \vdots \quad \vdots$$

$$X_5 + X_1 \xrightarrow{k_8^0} \text{decomposition products} + (\mathrm{GaAs})_{\mathrm{surf}} \tag{3.14}$$

where X_i are the decomposition intermediates, k_i^0 are the reaction rate constants, and $(\mathrm{GaAs})_{\mathrm{surf}}$ is a surface GaAs unit. The stabilization reaction is given by

$$Fe^{2+} + X_1 \xrightarrow{k_{-1}^s} Fe^{3+} + (\mathrm{GaAs})_{\mathrm{surf}}\,. \tag{3.15}$$

The analysis of their data for the case where surface recombination is important suggests that "ordinary" surface states (not originating from decomposition intermediates), higher-order decomposition intermediates (assumed to be X_2), and, to a small extent, X_1 are involved in the recombination process. These recombination steps are given by

$$R^0 + h^+ \xrightarrow{\beta_p} R^+ \tag{3.16}$$

$$R^+ + e^- \xrightarrow{\beta_n} R^0 \tag{3.17}$$

$$X_2 + e^- \xrightarrow{k_{-2}^r} X_1 \tag{3.18}$$

$$X_1 + e^- \xrightarrow{k_{-1}^r} (\mathrm{GaAs})_{\mathrm{surf}} \tag{3.19}$$

where R^0 and R^+ are the filled and empty states, respectively, β_p and β_n are capture rates, and k^r_{-2} and k^r_{-1} are reaction rates. For given values of the semiconductor parameters, photocurrent density i, and Fe^{2++} concentration y, the rate equations appropriate for the reactions given above can be simplified to yield

$$n_s = \frac{k^0_3}{k^r_{-2} \cdot k^s_{-1}} \cdot \frac{s \cdot j}{y} \cdot \left[\frac{6 \cdot k^0_2}{(k^s_{-1})^2} \cdot \frac{s^2}{1-s} \cdot \frac{j}{y^2} - 1 \right] \tag{3.20}$$

where $j = i/e_o$. It can be seen from Equation (3.20) that the rate law observed for the experimental results at constant voltage and varying light intensity is obtained when $n_s \simeq 0$. The reaction scheme proposed by Vanmaekelbergh *et al.* [85] is illustrated in Figure 3.7.

3.2 Experimental Methods

The rotating ring–disk electrode (RRDE) has been employed extensively to study the corrosion of semiconductors. The use and advantages of this experimental apparatus are reviewed in this section. The experimental objective and procedures are also discussed.

3.2.1 Objective of the Study

The overall objective of this work was to investigate the influence of redox species and light intensity on the corrosion of n-GaAs photoanode. To support this objective, hole–injection and stabilization experiments were performed using GaAs-disk Pt-ring electrodes. The etching rates were enhanced for hole–injection experiments using ferricyanide as the oxidizing agent. The stabilization of the photocorrosion processes was investigated with Fe^{2+} as the reducing agent. For this case, the potential range of interest corresponds to the onset of photocurrent and the light-limited current plateau.

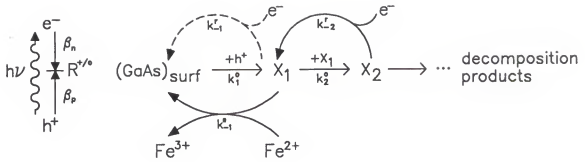


Figure 3.7: A reaction scheme for the decomposition, stabilization, and surface recombination processes for the GaAs/Fe^{2+} photoanode proposed by Vanmaekelbergh *et al.* [85]. Reprinted by permission of the publisher, *Berichte der Bunsengesellschaft für Physikalische Chemie.*

3.2.2 Rotating Planar Electrodes

The rotating disk electrode (RDE) and the RRDE are important analytical tools in the study of electrode kinetics. Levich [94] and Riddiford [95] have presented excellent reviews on the hydrodynamics and mass–transfer limited behavior of the RDE system. Newman [96] calculated theoretically the current distribution on an RDE below the limiting current, and his results have been observed experimentally [97, 98]. The theory and application of the RRDE were discussed in a series of publications [99–114].

3.2.2.1 The rotating disk electrode

The derivation of the Levich equation involves the solution of the hydrodynamic and convective–diffusion problems for a submerged disk electrode. The velocity distributions of the fluid motion can be expressed as series functions [94, 95]. Near the electrode surface, the radial velocity v_r and normal velocity v_z are approximated by the first term of series series given by

$$v_r = 0.510 r z \Omega^{3/2} \nu^{-1/2} \tag{3.21}$$

$$v_z = -0.510 z^2 \Omega^{3/2} \nu^{-1/2} \tag{3.22}$$

where Ω is the rotation speed, ν is the kinematic viscosity of the solution, and r and z are the radial and normal distances, respectively. This is valid for solutions of high Schmidt numbers ($N_{Sc} = \nu/D_i > 50$), where D_i is the diffusion coefficient of the electroactive species) because the diffusion layer is much smaller than the hydrodynamic layer and, for dilute solutions, the mass–transfer process does not perturb the hydrodynamic conditions [97].

The steady–state convective–diffusion equation can be simplified by neglecting electric migration and radial diffusion [94, 97, 100, 115]. These assumptions apply

for solutions with an excess supporting electrolyte and when mass-transport by convection is significantly greater than radial diffusion as in the case of a rotating disk electrode. The convective–diffusion equation reduces to a form given by

$$r\frac{\partial c_i}{\partial r} - z\frac{\partial c_i}{\partial z} = \frac{D_i}{0.510\Omega^{3/2}\nu^{-1/2}}\frac{1}{z}\frac{\partial^2 c_i}{\partial z^2}, \tag{3.23}$$

where c_i is the reactant concentration. The analysis is further simplified by the uniform accessibility condition, that is, the concentration of the reactant is a function only of the normal distance from the electrode such that radial gradients can be neglected ($\partial c_i/\partial r = 0$) [115]. This simplifies Equation (3.23) to an ordinary differential equation. Under mass–transfer limited conditions, the electroactive species is completely consumed at the disk. The appropriate boundary conditions are given by

$$c_i = 0 \quad \text{at} \quad z = 0, \quad \text{and} \tag{3.24}$$

$$c_i = c_\infty \quad \text{at} \quad z = \infty \tag{3.25}$$

where c_∞ is the bulk concentration of the reactant. The normal derivative of concentration and the reaction rate is related by Faraday's and Fick's laws such that

$$I_d = nF\pi r_1^2 D_i\left(\frac{\partial c_i}{\partial z}\right)_{z=0} \tag{3.26}$$

where I_d is the disk current, F is Faraday's constant, and r_1 is the radius of the disk electrode. The solution of Equation (3.23) at mass–transfer limited condition yields the Levich equation

$$I_{d,lim} = 0.62nF\pi r_1^2 D_i^{2/3}\nu^{-1/6}\Omega^{1/2}c_\infty \tag{3.27}$$

An analogous expression can be derived for a ring–electrode [94]

$$I^o_{r,lim} = 0.62nF\pi(r_3^3 - r_2^3)^{2/3}D_i^{2/3}\nu^{-1/6}\Omega^{1/2}c_\infty \tag{3.28}$$

where r_2 and r_3 are the inner and outer radius, respectively. $I^o_{r,lim}$ also denotes the limiting current at a ring electrode of an RRDE when the current at disk electrode

is zero; this value is also called the "unshielded" ring current. These equations show that the limiting currents at the electrodes should be a linear function of the square root of rotation speed.

Gregory [95] and Gregory and Riddiford [116] proposed a correction to Levich's equation to account for the differences observed between experimental measurements and predicted values for electrolyte solutions of low Schmidt numbers. They stated that second term of the series function for the normal velocity, which accounts for the Schmidt number, contributes significantly to the integral equation for lower values of N_{Sc}. The correction to the mass–transfer limited current at the disk electrode is given by

$$I_{d,lim} = \frac{0.554}{0.8934 + 0.316(D/\nu)^{0.36}} nF\pi r_1^2 D^{2/3}\nu^{-1/6}\Omega^{1/2}c_\infty \qquad (3.29)$$

and provided excellent correlation between the experimental and predicted values of $I_{d,lim}$ for Schmidt numbers less than 250 [116].

3.2.2.2 The rotating ring–disk electrode

The ring–disk electrode is a versatile tool for studying reversible redox reactions. The bipotentiostatic operation of the RRDE permits independent control of the disk and ring electrodes. Consequently, an electroactive species generated (or consumed) at the disk can be detected by changes in the ring current. Collection efficiency and shielding experiments are two common applications of the RRDE, and illustrative examples for the Cu(II)/Cu(I)/Cu system are given in Reference [117]. Consider the reversible reaction

$$A \rightleftharpoons B + ne^- . \qquad (3.30)$$

The anodic formation of species B at the disk can be detected at the ring electrode being held at cathodic potential. The ratio of the ring current to the disk current, when mass–transfer limited conditions apply at both electrodes, is the collection

efficiency N_o given by

$$N_o = -\frac{I_{r,lim}}{I_{d,lim}}. \tag{3.31}$$

When both the disk and ring electrodes are held at anodic potentials simultaneously, the RRDE acts as a sectioned–disk electrode. The disk electrode effectively "shields" the ring electrode from reactant A which would otherwise have reached the ring. Relationship between the "unshielded" limiting current at the ring $I^o_{r,lim}$ and the limiting current at the disk $I_{d,lim}$ is a function only of the geometric design of the RRDE. From Equations (3.27) and (3.28), the unshielded ring current is given by [101]

$$\begin{aligned} I^o_{r,lim} &= \left[\left(\frac{r_3}{r_1}\right)^3 - \left(\frac{r_2}{r_1}\right)^3\right]^{2/3} I_{d,lim} \\ &= \beta^{2/3} I_{d,lim} \end{aligned} \tag{3.32}$$

where β is a geometric parameter. The ratio of the shielded ring current to the unshielded ring current is the shielding efficiency S_o. For a properly constructed RRDE, it has been shown that the collection efficiency and the shielding efficiency are functions only of the electrode geometry.

Collection efficiency. Albery and Bruckenstein [100] derived an exact expression for the collection efficiency by solving the convective–diffusion equation (3.23) subject to boundary conditions applicable to the various zones of the RRDE. The collection efficiency N_o is a function only of the electrode geometry given by

$$\begin{aligned} N_o = {} & 1 - F(\alpha/\beta) + \beta^{2/3}[1 - F(\alpha)] - \\ & (1+\alpha+\beta)^{2/3}\{1 - F[(\alpha/\beta)(1+\alpha+\beta)]\} \end{aligned} \tag{3.33}$$

where $\alpha = (r_2/r_1)^3 - 1$, β is the geometric parameter defined in Equation (3.32), and

$$\begin{aligned} F(\theta) &= \frac{\sqrt{3}}{2\pi}\int_0^\theta \frac{d\lambda}{\lambda^{2/3}(1+\lambda)} \\ &= \frac{\sqrt{3}}{4\pi}\ln\frac{(1+\theta^{1/3})^3}{1+\theta} + \frac{3}{2\pi}\arctan\left(\frac{2\theta^{1/3}-1}{\sqrt{3}}\right) + \frac{1}{4}. \end{aligned} \tag{3.34}$$

These authors have tabulated the values of N_o for various RRDE geometries in Reference [100]. Smyrl and Newman [97] verified the derivation of Albery using another mathematical approach. The experimental results reported by Albery *et al.* [100] and Napp *et al.* [117] agree with the predicted values given by Equation (3.33).

Shielding efficiency. For shielding experiments, the ring and disk electrodes are at mass-transfer limited conditions for the same reaction. The "unshielded" ring current is decreased by an amount $N_o \cdot I_{d,lim}$ due to the consumption of the reactant at the disk. Therefore, the shielded ring current is given by [101, 117]

$$I_r = (\beta^{2/3} - N_o) I_{d,lim}. \tag{3.35}$$

From Equations (3.32) and (3.35), the shielding efficiency is [117]

$$S_o = I_r / I^o_{r,lim} = 1 - N_o/\beta^{2/3}. \tag{3.36}$$

Effects of current distribution. The equation for the collection efficiency of a RRDE was derived under the assumption that the concentration of the generated species at the disk is uniform. This assumption will be valid when the disk is at mass-transfer limited condition because the current distribution at the disk electrode is nearly uniform [96]. Below the limiting current, however, a non-uniform current distribution exist due to the influence of surface overpotential and ohmic drop. Albery and Ulstrup [118] suggested using the RRDE to test Newman's theory. They stated that a non-uniform current distribution at the disk will result in a higher value of the collection efficiency as compared to that predicted by Equation (3.33); however, their initial results had discrepancies with Newman's numerical results [118]. Later work [97, 98, 114] showed that higher collection efficiencies are indeed obtained below the limiting current, which implies higher concentration gradients at the disk edge. Smyrl and Newman [97] also suggested that use of the RRDE as a sectioned-disk below the limiting current would have the advantage of making the current

distribution at the disk electrode more uniform since the ring is effectively "'soaking up' the nonuniformities."

3.2.3 Electrode Fabrication

The RRDE employed for this study was a 0.50 cm GaAs disk and a platinum ring assembly. The Pt–ring assemblies were obtained from Pine Instrument Company[1] and have inner- and outer-diameters of 0.750 ± 0.002 cm and 0.850 ± 0.002 cm, respectively. These dimensions provide a theoretical collection efficiency N_o of 0.225, geometric parameter $\beta^{2/3}$ of 1.332, and a theoretical shielding efficiency S_o of 0.831 using Equations (3.33), (3.32) and (3.36), respectively.

The semiconductor disk electrodes were fabricated using n-type silicon–doped GaAs wafers manufactured by Laser Diode.[2] The specifications of the semiconductor material provided by the manufacturer are as follows: resistivity $\rho = 0.0015$–0.0026 Ω-cm, mobility $\mu = 1164$–2025 cm^2/V-s, and doping density $N = 1.2 \times 10^{18}$–2.5×10^{18} cm^{-3}. The wafers were 17 mils (0.0432 cm) thick.

The 0.5 cm diameter GaAs disks were sliced from the wafers using a Gatan[3] Model 601 Ultrasonic Disk Cutter and a 0.5 cm cutting die. Electrical contact was formed on one side of the semiconductor disk by alloying 99.999% indium at 400°C for 10 minutes [85] under an inert H_2–N_2 flowing gas stream. The GaAs disk was mounted on a stainless shaft which had a cavity filled with conductive silver paste.

The electrodes purchased from Pine Instruments Company had a dismountable configuration. A schematic diagram of the electrode is presented in Figure 3.8. It consisted of a Teflon shrouded Pt–ring assembly and an inner teflon U–cup shroud for

[1]Pine Instrument Company, Grove City, PA 16127.

[2]Laser Diode, Inc., 4014 Wedgeway Court, Earth City, MO 63045.

[3]Gatan, Incorporated, 780 Commonwealth Drive, Warrendale, PA 15086.

the GaAs-disk/stainless steel shaft unit. The steel shaft was secured with a nut on the end protruding from the bottom of the U-cup. The disk assembly was press-fitted into the Pt-ring assembly to form the RRDE. The entire assembly was attached to a Pine Instrument rotator mounting shaft designed for bi-potentiostatic operation.

A primary concern was the fabrication of an RRDE that would be free of extraneous currents. Preliminary experiments showed that electrolyte seeped in between the teflon U-cup and the ring- and disk-assemblies due to capillary action. This caused corrosion of the Pt-ring brass shaft and noisy current-voltage measurements. The problem was resolved by laminating the inside of the ring assembly and the GaAs/steel shaft assembly with resin epoxy.

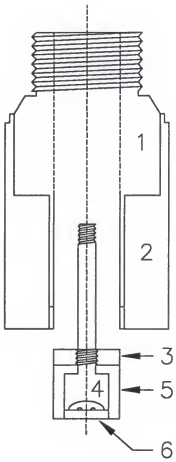
3.2.4 Experimental Procedure

A schematic diagram of the experimental setup is presented in Figure 3.9. The electrolyte beaker had two side-arms to house the platinum counterelectrode and the reference electrode. The reference electrode used were the saturated calomel electrode (SCE) and the double-junction silver-silver chloride electrode (Ag/AgCl). The bottom of the beaker had an optically flat microscope mounting glass to permit illumination of the GaAs disk from below using a Dolan-Jenner[4] Fiber-Lite. The light was directed using fiber-optic cable arms with focusing lenses, and the intensity was controlled using neutral density filters obtained from Melles-Griot.[5]

The electrolyte solutions were prepared from reagent grade chemicals (Fe_2SO_4, Na_2SO_4, KOH, $K_3Fe(CN)_6$, and $K_4Fe(CN)_6$) and deionized water. The pH of the solution was adjusted using 98% w/w sulfuric acid. Reduction of oxygen can take place at the potential used in this study and could obscure the measurement of redox

[4]Dolan-Jenner Industries, Incorporated, Blueberry Hill Road, Box 1020, Woburn, MA 01801.

[5]Melles-Griot, 1770 Kettering St., Irvine, CA 92714.



Ring Assembly:

1: Pt–ring brass shaft
2: Teflon shroud

Disk Assembly:

3: Nut
4: Stainless steel shaft
5: Teflon U–cup
6: GaAs disk

Figure 3.8: Schematic diagram of the GaAs–disk platinum–ring assembly. The diameter of the disk electrode is 0.5 cm, and the inner and outer diameters of the platinum ring electrode are 0.75 cm and 0.85 cm, respectively.

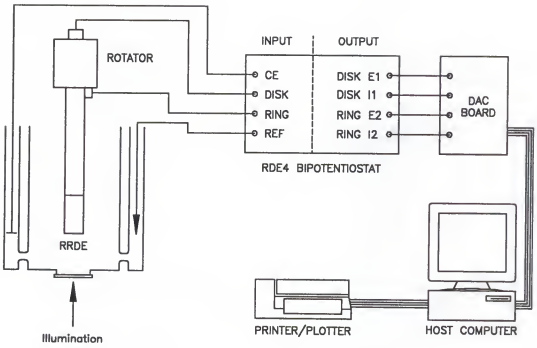


Figure 3.9: Schematic diagram of the experimental setup.

currents. Thorough deaeration of the solution was therefore critical to the success of these experiments. The solutions were deaerated by placing the prepared electrolyte under vacuum for three to five hours followed by argon gas bubbling for several more hours. The elimination of oxygen was confirmed for a 0.25M Na_2SO_4, pH = 1.44 "blank" solution which did not contain any redox species. The potential of a platinum ring electrode was held at -0.2V (Ag/AgCl) and the current was measured as a function of rotation speed. The measured cathodic currents were less than 3.5 μA and showed no linear dependence with rotation speed. This value is negligible when compared to the mass–transfer limited current observed at the ring electrode for the reduction of ferric at this potential.

The GaAs disk electrodes were pre-etched in 0.1M KOH and 0.15M $Fe(CN)_6^3$ solution under illumination for 3 to 5 minutes until a shiny surface was obtained. It was necessary to re-etch the semiconductor disk electrode after several runs, especially when illuminated at high intensities of long duration. The usual procedure of pre-etching the GaAs samples with a 98% H_2SO_4::30%w H_2O_2 (1::1) solution was not employed because it damaged the epoxy resin laminate of the disk and ring assemblies.

The RRDE assembly described in the previous section was attached to a Pine Instrument Model ASR Analytical Rotator which provides electrical contacts for the ring and disk electrodes. A Pine Instrument RDE4 Analog Bipotentiostat was used for the cyclic voltammetry and potentiostatic experiments. The bipotentiostat allows independent control of the disk (potentiostatic, potentiodynamic, or galvanostatic mode) and ring (potentiostatic mode only) electrodes. Open circuit measurements were performed on a Princeton Applied Research[6] (PAR) Model 273 Potentiostat.

[6]EG&G Princeton Applied Research, P.O. Box 2565, Princeton, NJ 18540.

Data acquisition was computer-controlled and interfaced using a Metrabyte[7] DAS-16F anolog-to-digital conversion (ADC) board or a Hewlett–Packard[8] 7090A digitizing plotter. The ADC boards have a 12-bit resolution and adjustable full–scale voltage input. The sensitivity of the ADC board was typically between 1.0 to 2.5 mV/bit. The data acquisition program was written in Borland[9] Turbo Pascal 5.5 which used the software driver provided by Metrabyte [119] or which implemented the HP-IB instructions for the HP-7090A plotter [120] and PAR potentiostat [121].

3.3 Results and Discussion

Preliminary experiments using a platinum-disk and platinum-ring electrode were performed to evaluate the technique of fabricating an RRDE. Experimental results for hole-injection experiments and stabilization experiments are discussed.

3.3.1 Platinum RRDE

The dismountable configuration of the RRDE obtained from Pine Instruments was tested using a platinum disk electrode. The electrolyte solution was 1.0M KOH and 0.010M $K_4Fe(CN)_6{\cdot}3H_2$. A potential scan from 0.0 V to 0.7V (*vs.* SCE) was applied to the platinum disk electrode or the platinum ring electrode. The limiting current for the anodic reaction

$$Fe(CN)_6^{-4} \longrightarrow Fe(CN)_6^{-3} + e^- \tag{3.37}$$

was measured as a function of rotation speed and is illustrated in Figure 3.10. Experimental values for the geometric factor $\beta^{2/3}$ can be determined using equation (3.32)

[7]Metrabyte Corporation, 440 Myles Standish Blvd., Taunton, MA 02780.

[8]Hewlett–Packard Company, 16399 W. Bernardo Drive, San Diego, CA 92127–1899.

[9]Borland International, 1800 Green Hills Road, Scotts Valley, CA 95067.

Table 3.1: Experimental values for the geometric parameter $\beta^{2/3} = I_{r,lim}/I_{d,lim}$.

Ω, RPM	$I_{r,lim}$, mA	$I_{d,lim}$, mA	$\beta^{2/3}$
400	0.73	0.57	1.28
750	1.00	0.77	1.30
1000	1.18	0.89	1.33
1250	1.31	1.00	1.31
1600	1.51	1.12	1.35

which relates the unshielded limiting current at the ring to the limiting current at the disk. The results are tabulated in Table 3.1. The averaged experimental value for $\beta^{2/3}$ is 1.31, which is within 2% of the theoretical value of 1.332.

The slope of the lines in Figure 3.10 correspond to the slope of the Levich equation, and were determined by least–square regression to be 8.70×10^{-5} and 11.5×10^{-5} $A/(rad/sec)^{1/2}$ for the disk and ring electrode, respectively. From equations (3.27) and (3.28), the slope applicable to the disk electrode is given by

$$Slope_{\text{disk}} = 0.62nF\pi R_1^2 D_i^{2/3}\nu^{-1/6}c_\infty \tag{3.38}$$

and for the ring electrode is given by

$$Slope_{\text{ring}} = 0.62nF\pi(R_3^3 - R_2^3)^{2/3}D_i^{2/3}\nu^{-1/6}c_\infty . \tag{3.39}$$

The density of the solution was 1.05 gm/cm^3, and the kinematic viscosity ν was estimated to be 8.6×10^{-3} cm^2/s, assuming that the solution's viscosity is that of water at 25°C ($\sim$ 0.90 cP). Since $c_\infty = 0.010M$ and the geometric parameters of the RRDE are known, the experimental value of the diffusion coefficient D_i of ferrocyanide $Fe(CN)_6^{-4}$ was found to be 0.61×10^{-5} cm^2/s. The experimental value is within 3% of the predicted value of 0.631×10^{-5} cm^2/s estimated from the correlation of Gordon

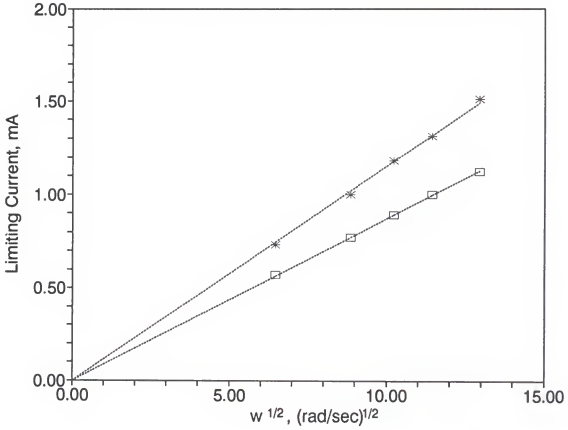


Figure 3.10: Limiting currents at the platinum ring (∗) and platinum disk electrode (□) as a function of rotation speed. The electrolyte solution was 1.0M KOH and 0.010M $K_4Fe(CN)_6 \cdot 3H_2O$.

et al. [122]. The measured collection efficiency at a 1000 RPM was found to be 0.21 as compared to the theoretical value of 0.225 (difference of 7%).

The results presented here were obtained with a new RRDE assembly and teflon U-cup. The RRDE employed for these experiments was essentially leak-free, and the experimental values obtained for $\beta^{2/3}$, N_o, and $D_{Fe(CN)_6^{-4}}$ are comparable to theoretical values in literature. It was later discovered that the snug fit of the teflon U-cup deteriorated with repeated removal of the disk assembly, causing a leakage current at the ring electrode due to corrosion of the inner brass shaft. The problem was resolved by laminating the inside of the ring electrode with epoxy resin. In addition, the outer Teflon shrouds of several ring assemblies were eventually replaced by casting the electrode in epoxy resin.

3.3.2 Hole Injection Experiments

The effect of hole injection on the anodic dissolution of n-GaAs is illustrated in this section. The oxidizing agent is ferricyanide, which has been described as an efficient hole injector [67]. Hole injection involves capture of electrons from the valence band by the reduction of ferricyanide at the GaAs disk electrode, *i.e.*,

$$Fe(CN)_6^{-3} \longrightarrow Fe(CN)_6^{-4} + h^+ . \tag{3.40}$$

The RRDE was a 5-mm n-GaAs disk and a platinum ring electrode as described in the previous section. The electrolyte solution was 0.10M KOH and 0.1M $K_3Fe(CN)_6$. The applied potential at the Pt-ring electrode was held constant at +0.6V (*vs.* Ag/AgCl) to detect the formation of ferrocyanide at the disk. Potentiodynamic scans at 20 mV/s were applied to the disk electrode using two rotation speeds (100 and 500 RPM) and varying light intensity. Neutral density filters were employed to control illumination. Light intensity is reported as percent transmission %T, with 100%T referring to the

case with no filters. The illumination intensity at the GaAs disk electrode could not be measured during the experiment, but the intensity at 100%T was estimated to be 5 mW/cm^2.

For a rotation speed of 100 RPM, the current–potential (I-V) curves for the ring and disk electrodes are shown in Figure 3.11 under dark and illuminated conditions. The constant ring current in the dark suggests that the rate of formation of ferrocyanide on the GaAs disk was constant over the entire potential range. The rate of hole–injection on the n–GaAs disk in the dark was constant and, therefore, limited by the transport of oxidizing agent to the GaAs disk. At cathodic potentials, the reduction in band–bending lead to a higher rate of electron–hole recombination and the injected holes recombined with the electrons in the space charge region. At anodic potentials, the injected holes accumulated at the surface due to a reduction in the recombination rate and contributed to the decomposition of GaAs. This is illustrated in Figure 3.11 by the anodic increase in the disk current at dark condition. Under illumination, the ring currents were constant at potentials more negative than $-0.6V$ (*vs.* Ag/AgCl) and decreased dramatically at more positive potentials. This meant that the formation of ferrocyanide, or, equivalently, the rate of hole injection, at the GaAs disk electrode was reduced at more anodic potentials. The enhanced band–bending at anodic potentials reduced the rate of recombination of electrons and holes at the surface. The accumulation of photogenerated holes at the semiconductor surface resulted in kinetic–limitation of the hole–injection process. These results are consistent with the experimental results presented by Notten [67]. Notten [67] reported that the maximum dissolution rate of a GaAs photoanode is determined by the pH of the solution and electrode rotation rate. In the dark and under low light intensities, when the rate of decomposition due to hole–injection and/or photogenerated holes is lower that the maximum dissolution rate, he showed that the rate of hole–injection is

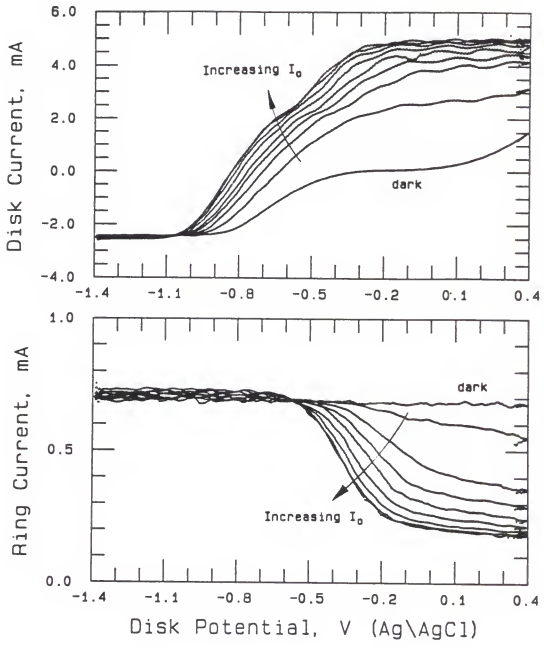


Figure 3.11: Current–potential curves for an n-GaAs disk and platinum ring electrode rotating at 100 RPM in 0.10M KOH and 0.10M $K_3Fe(CN)_6$ solution.

diffusion–limited over the entire potential sweep range. The photogeneration of holes is influenced by illumination, while the rate of hole–injection is affected by rotation speed. When the sum of the rates of these two processes exceeds the maximum dissolution rate, or under intense illumination alone, the rate of hole–injection becomes kinetically controlled at anodic potentials.

Since the rate of hole injection is influenced by rotation speed, similar experiments were performed at 500 RPM. The current–potential curves under illuminated conditions are presented in Figure 3.12. Kinetic limitation to the hole–injection process was also observed as shown by the decreasing ring currents at potentials greater than -0.9V (Ag/AgCl). However, this effect was less drastic at 500 RPM as compared to a rotation speed of 100 RPM under similar conditions of illumination due to a higher flux of ferricyanide. Higher photocurrents were measured which suggests that the rate of photocorrosion also increased at 500 RPM due to a higher flux of hydroxyl ions, a reactive species in the decomposition of GaAs [61, 65, 67]. The photocurrent plateaus were found to be proportional to light intensity except at the highest intensities, as shown in Figure 3.13.

The partial anodic and cathodic currents on the GaAs disk electrode, due to the decomposition of the semiconductor and the reduction of ferricyanide, respectively, were determined from the I–V curves described above. A value for the collection efficiency of the RRDE was required in order to deconvolute the contributions of the partial reactions to the total disk current. The averaged disk currents at cathodic potentials were determined to be -2.5 ± 0.1 mA and -5.8 ± 0.1 mA for 100 RPM and 500 RPM, respectively. The corresponding averaged ring currents were found to be 0.7 ± 0.03 mA and 1.51 ± 0.04 mA for 100 RPM and 500 RPM, respectively. Since data were available for two rotation rates only, Levich's equation was used to determine if the averaged current plateaus corresponded to mass-transfer limited conditions. The

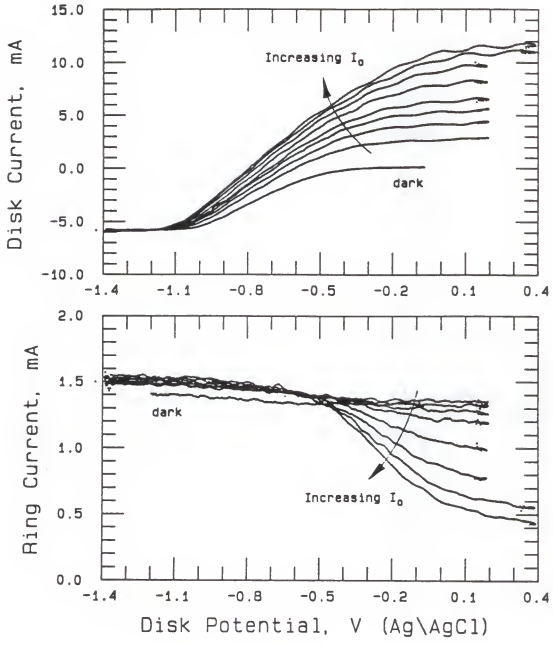


Figure 3.12: Current–potential curves for an n-GaAs disk and platinum ring electrode rotating at 500 RPM in 0.10M KOH and 0.10M $K_3Fe(CN)_6$ solution.

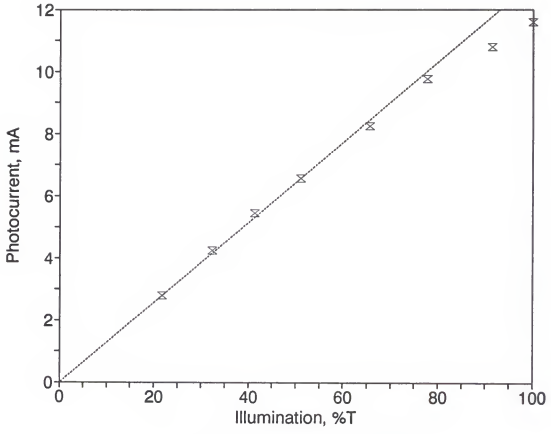


Figure 3.13: Photocurrent as a function of relative light intensity (percent transmission) at the n-GaAs disk–Pt ring electrode described in Figure 3.12.

relationship between the currents at two rotation rates is given by

$$I_{lim,500RPM} = \sqrt{5}\, I_{lim,100RPM} \,. \tag{3.41}$$

The experimental values were found to agree with the predicted values to within 4 percent. Consequently, the reactions at the ring and disk electrode were judged to be under mass–transfer limitation.

Using these limiting current values, the experimental collection efficiencies were estimated to be 0.28 and 0.26 for 100 RPM and 500 RPM, respectively. (It will be noted here that the experimental runs at 500 RPM were performed before the experimental runs at 100 RPM.) The estimated collection efficiencies deviate significantly from the theoretical value of 0.225, *i.e.,* by as much 24% and 16%, respectively. The I–V curves reported here were obtained over a period of one hour without changing the solution. It is possible that ferrocyanide, which was not collected at the ring electrode, accumulated in the bulk. This will increase the limiting current measured at the ring electrode over time. It may be possible that extended etching of the GaAs disk increased the effective surface area of the disk electrode or the reactivity of the semiconductor resulting in higher currents. The first experimental run was performed at 500 RPM under dark conditions; the disk current and ring current are shown in Figure 3.12. The disk current at cathodic potentials and the corresponding ring current were found to be −5.8 mA and 1.4 mA, respectively, at the start of the experiment. The 0.1 mA difference between ring currents measured at 500 RPM suggest that ferrocyanide accumulated in the bulk and contributed a "background" current. Furthermore, the initial collection efficiency of 0.24 ($= -I_r/I_d = 1.4$ mA/5.8 mA) is closer to the theoretical value.

Since independent measurements of the "background" anodic currents at the ring electrode were not available, a baseline value of 0.1 mA was used to correct the ring current at 500 RPM. A baseline value of 0.07 mA ($= 0.7 - 1.4/\sqrt{5}$) was employed for

the ring current at 100 RPM. The reduction of ferricyanide at the disk was estimated from the baseline–adjusted ring data using equation (3.31) and a collection efficiency of 0.24. The partial anodic and cathodic currents at the semiconductor disk electrode are illustrated in Figures 3.14 and 3.15 for 100 RPM and 500 RPM, respectively. At a rotation speed of 100 RPM, the anodic decomposition attains a maximum value of approximately 5.5 mA which is independent of increasing light intensity as shown in Figure 3.14. These results show that the rate of dissolution by photogenerated holes is limited by the maximum rate of dissolution at the pH of the solution as reported by Notten [67]. The reaction mechanism for the decomposition of GaAs in basic solution involves hydroxyl ions [65]. Limitations to the anodic dissolution rate suggest a mechanism for the formation of an oxide layer and its subsequent dissolution by hydroxyl ions [67]. When the electrode rotation rate was increased, the flux of OH^- to the GaAs surface was increased. Consequently, higher anodic currents were observed at the GaAs disk electrode for a rotation rate of 500 RPM as illustrated in Figure 3.15. There was minimal limitation to the hole–injection process at 500 RPM because the injected holes which accumulated at the semiconductor surface at more anodic potentials contributed to the decomposition of the substrate.

3.3.3 Stabilization Experiments

Stabilization experiments were performed for an n-type GaAs-disk Pt-ring electrode immersed in a solution containing 0.5M or 1.0M Fe^{+2} in 0.25M Na_2SO_4, with the pH adjusted to 1.4 using sulfuric acid. The solutions were deaerated following the procedure described in the previous section. The potential of the ring electrode was held at -0.2V (Ag/AgCl). A non-negligible concentration of ferric ions was detected at cathodic potentials, even for a fresh solution. This was due to the oxidation of ferrous sulfate by oxygen during the preparation of the solution.

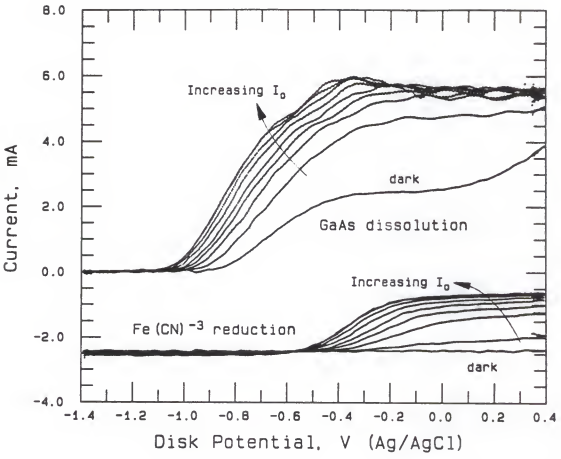


Figure 3.14: Partial anodic and cathodic current curves at the n-GaAs disk electrode, rotating at 100 RPM, calculated from the I–V curves shown in Figure 3.11.

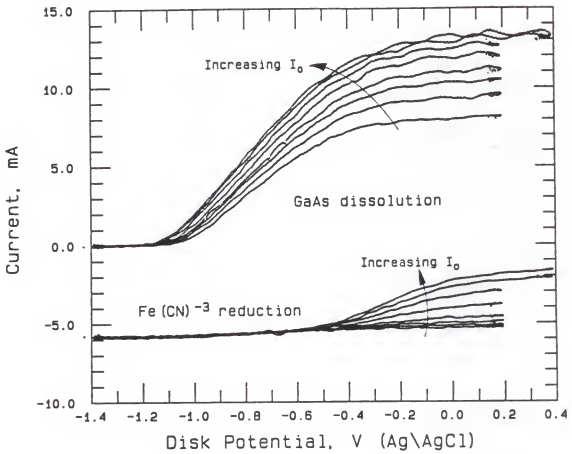


Figure 3.15: Partial anodic current curves at the n-GaAs disk electrode, rotating at 500 RPM, calculated from the I–V curves shown in Figure 3.12.

The collection efficiency N_o and the geometric parameter $\beta^{2/3}$ were calculated from the mass-transfer limited cathodic currents at the disk and ring electrodes. In this experiment, reduction of ferric ions occurs at both the ring and disk electrodes. Therefore, the disk effectively shields the ring electrode by consuming some of the Fe^{+3} that would have otherwise been consumed at the ring. The unshielded ring current was measured independently using another Pt-ring electrode with no disk present. An illustrative example is shown in Figure 3.16 for an RRDE rotating at 1000 RPM. At cathodic potentials, the averaged disk current and the shielded ring current were measured to be -0.115 mA and -0.120 mA, respectively. Using equation (3.35), the shielding factor ($\beta^{2/3} - N_o$) was calculated to be 1.044 ($= I_r/I_{d,lim} = 0.120/0.115$). The unshielded ring current, obtained independently using a ring electrode without a disk present, is shown in Figure 3.16 as the dashed line and was measured to be -0.141 mA. Using equation (3.32), the geometric factor $\beta^{2/3}$ was calculated to 1.226 ($= I^o_{r,lim}/I_{d,lim} = 0.141/0.115$). Therefore, the collection efficiency for this electrode was estimated to be 0.182 ($= N_o = 1.226$ - 1.044).

3.3.3.1 Potentiodynamic experiments

Potentiodynamic experiments were performed in 0.5M Fe^{+2}, 0.25M Na_2SO_4 solution (pH = 1.4) at several rotation rates and at two light intensities. The purpose of these experiments was to determine whether stabilization can be observed at the specified light intensities. The potential at the GaAs disk was scanned from -0.6V to $+0.2$V (Ag/AgCl) at a sweeprate of 10 mV/s. The ring electrode was held at a constant potential of -0.2V (Ag/AgCl) to monitor the flux of Fe^{+3}. From potentiodynamic scans under dark conditions, the averaged shielding factor ($\beta^{2/3} - N_o$) for this electrode was determined to be 1.00 ± 0.05. The deviation from the theoretical value of 1.10 was attributed to the imperfect planar surface of the dismountable RRDE.

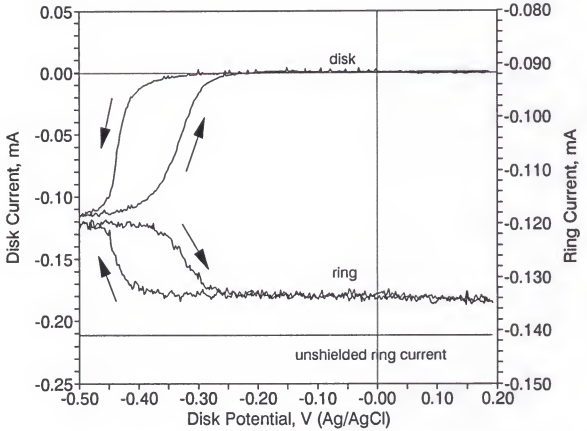


Figure 3.16: Potentiodynamic scan for an n-GaAs disk Pt-ring electrode under dark conditions rotating at 1000 RPM in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution. The ring electrode was held at a constant potential of -0.2V (Ag/AgCl). Solid line: RRDE I-V curves. Dashed line: Ring electrode limiting current.

The intensity of the light was varied using neutral density filters. The two cases presented here are for relative light intensities corresponding to 50% transmission (%T) and 100%T.

The current–potential curves for a GaAs-disk and Pt-ring electrode rotating at 500 RPM are shown in Figures 3.17 and 3.18 for light intensities of 50%T and 100%T, respectively. The hysteresis under illuminated condition depended on rotation speed and is characteristic of time-dependent behavior. The values of the unshielded ring currents $I^o_{r,lim}$ were estimated using equation (3.36), with the experimental shielding factor $(\beta^{2/3} - N_o) = 1.00$ and $N_o = 0.182$. The dashed lines in Figures 3.17 and 3.18 (and in Figures C.1 to C.12) correspond to the unshielded ring currents. It will be shown below that ferric reduction at the ring electrode at cathodic potentials is at mass-transfer limited conditions. Additional current–potential curves were obtained at rotation speeds of 150, 300, 750, 1000, and 1300 RPM. These I-V curves are presented in Appendix C. The potentiodynamic scans at 50% light transmission and at a given rotation rate are illustrated in Figures C.1 to C.6, while those for the 100%T are shown in Figures C.7 to C.12.

For each potential scan and rotation speed, the averaged disk current and the corresponding ring current were measured in the region of large cathodic potentials, typically from -0.6V to -0.5V. An averaged ring currents were also determined corresponding to the potential region of light-limited currents at the disk. These values are presented in Figures 3.19 and 3.20 for the 50%T and 100%T, respectively, as a function of the square root of rotation speed. The linear dependence of the disk and ring currents with $\Omega^{1/2}$ show that the reactions at the ring and disk electrodes are mass-transfer limited for the specified potential ranges.

At cathodic potentials and under illumination, the limiting current at the GaAs disk was more positive than those measured in the dark under similar conditions.

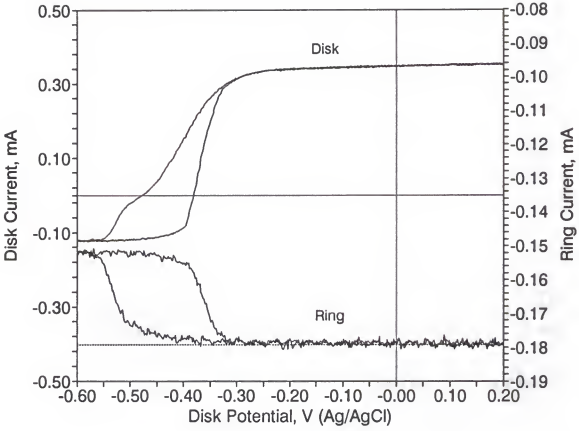


Figure 3.17: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 500 RPM in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution. The relative light intensity was 50%T. The ring electrode was held at a constant potential of -0.2V (Ag/AgCl). Solid line: RRDE I-V curves. Dashed line: Unshielded ring-electrode limiting current.

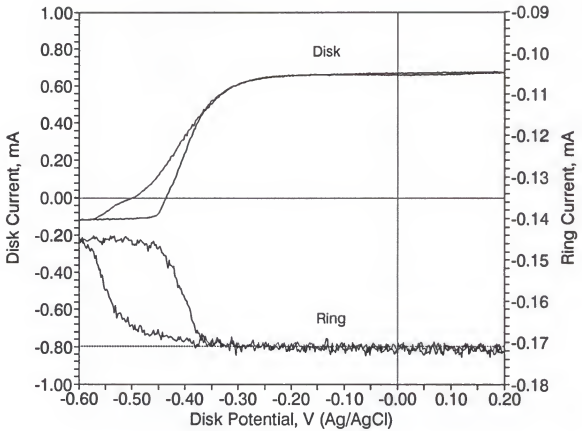


Figure 3.18: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 500 RPM in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution. The relative light intensity was 100%T. The ring electrode was held at a constant potential of -0.2V (Ag/AgCl). Solid line: RRDE I-V curves. Dashed line: Unshielded ring-electrode limiting current.

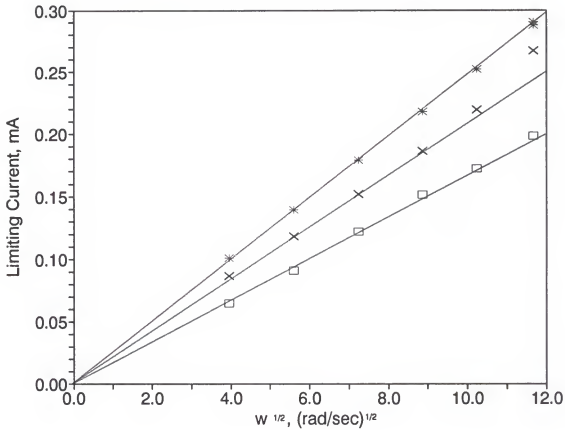


Figure 3.19: Averaged disk and ring currents obtained from Figures C.1 to C.6 for the 50% light transmission as a function of the square root of rotation frequency. □, ×: averaged disk and corresponding ring current at cathodic potentials. ∗: averaged ring current corresponding to the potential region of light-limited current at the disk.

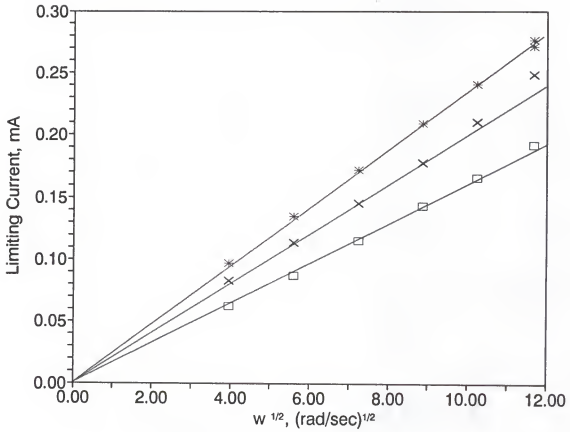


Figure 3.20: Averaged disk and ring currents obtained from Figures C.7 to C.12 for the 100% light transmission as a function of the square root of rotation frequency. □, ×: averaged disk and corresponding ring current at cathodic potentials. ∗: averaged ring current corresponding to the potential region of light-limited current at the disk.

This suggests that the illuminated GaAs disk was photocorroding at these potentials, since the total current at the disk would be the sum of the ferric reduction and an anodic reaction. At more anodic potentials, the disk current attained a light-limited value independent of rotation speed. The corresponding ring current can be compared to the unshielded ring current $I^{o}_{r,lim}$. For both light intensities with rotation speeds below 1000 RPM, the ring currents reverted to the unshielded values at the onset of photocurrent as shown in Figures 3.17 and 3.18. This means that ferrous oxidation was not occurring at the GaAs disk electrode since additional formation of ferric was not detected at the ring electrode beyond the unshielded current value. Therefore, the observed photocurrent was due primarily to the decomposition of the semiconductor. The photoanode was not stabilized and ferrous oxidation could not compete kinetically with the GaAs photodecomposition reaction at these light intensities. At 1000 and 1300 RPM, the ring currents did not revert to the calculated unshielded ring currents. This means that ferric was still being reduced at more positive potentials. Since higher ring currents were observed at the cathodic potential region when compared to the $\Omega^{1/2}$ dependence in Figures 3.19 and 3.20, the increased agitation may have increased the oxygen content of the solution inspite of the argon gas blanket leading to the homogeneous oxidation of the ferrous ions. This is apparent in the discrepancy between the ring-electrode measurements for the forward and reverse sweeps at higher rotation frequencies.

These experiments showed that the presence of 0.5M Fe^{+2} could not stabilize the GaAs photoanode and could not compete with the dissolution reaction at light intensities of 50%T and 100%T. Furthermore, the rotation speed has to set below 1000 RPM to minimize incorporation of oxygen into the solution.

3.3.3.2 Potentiostatic experiments

Potential step experiments were performed to determine the stabilization of the photodecomposition reaction as a function of applied potential. Based on the results presented in the previous section, light transmission was reduced to 20%T and the rotation speed of the electrode was fixed at 500 RPM to minimize aeration of the solution. The electrolyte solution was 0.5M Fe^{2+}, 0.25M Na_2SO_4, and pH = 1.4. The ring potential was held constant at -0.2V (Ag/AgCl) to monitor the flux of Fe^{+3}. The disk current was measured at constant potential under dark and illuminated conditions.

The experimental procedure was designed to be partially self-calibrating. At the start of the experiment, the disk potential was held at -0.6V (Ag/AgCl) for several minutes without illumination in order to determine a value for the shielding factor. In the dark and at -0.6V (Ag/AgCl), the reaction at the GaAs disk will be the reduction of ferric ions present in solution. The disk electrode shields the ring electrode by reducing the ferric ions which would have otherwise reached the ring electrode. The unshielded ring currents were estimated from equation 3.36 using a known value of collection efficiency. The disk potential was then stepped to the specified potential without illumination for several more minutes. This procedure was expected to distinguish between the generation (*i.e.*, collection) or the consumption (*i.e.*, shielding) of ferric ions in the dark while the disk electrode was held at the specified potential. The electrode was then illuminated for the duration of the experimental run at the specified potential. A ring current greater than the unshielded current value corresponds to oxidation of Fe^{+2} at the disk electrode and stabilization of photocorrosion.

Currents transients for potential steps to $+0.1$V, -0.25V and -0.35V are shown in Figures 3.21, 3.22, and 3.23, respectively. For each case, the shielding factor ($\beta^{2/3} - N_o$) was estimated from the averaged disk and ring currents at -0.6V

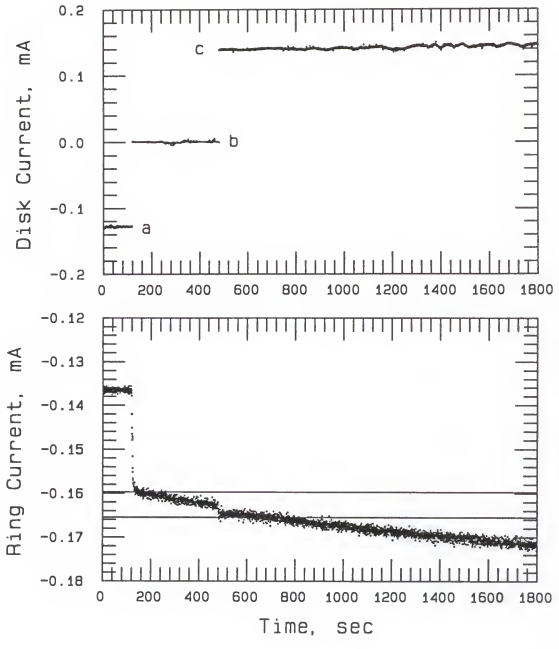


Figure 3.21: Disk and ring currents for the potential step experiment (+0.1V (Ag/AgCl)) of an n-GaAs-disk/Pt-ring electrode in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution, with a rotation speed of 500 RPM. The relative light intensity was 20%T, and the potential of the ring electrode was held constant at −0.2V. a) $I_d = -0.6$V (Ag/AgCl) without illumination. b) $I_d = +0.1$V (Ag/AgCl) without illumination. c) $I_d = +0.1$V (Ag/AgCl) with illumination.

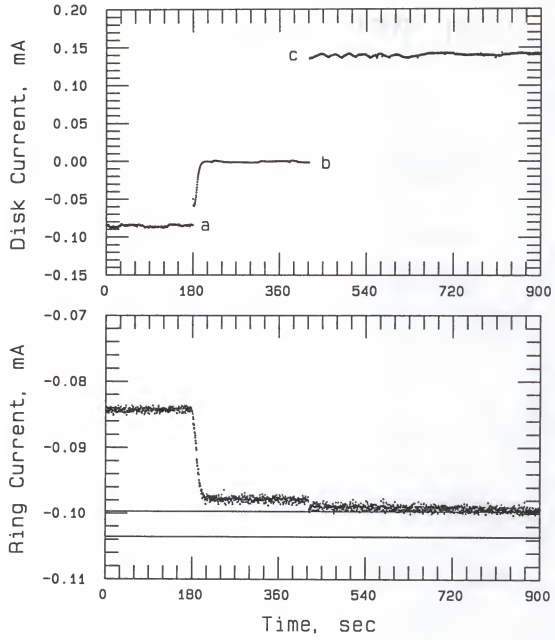


Figure 3.22: Disk and ring currents for the potential step experiment (−0.25V (Ag/AgCl)) using same experimental system described in Figure 3.21. a) $I_d = -0.6$V (Ag/AgCl) without illumination. b) $I_d = -0.25$V (Ag/AgCl) without illumination. c) $I_d = -0.25$V (Ag/AgCl) with illumination.

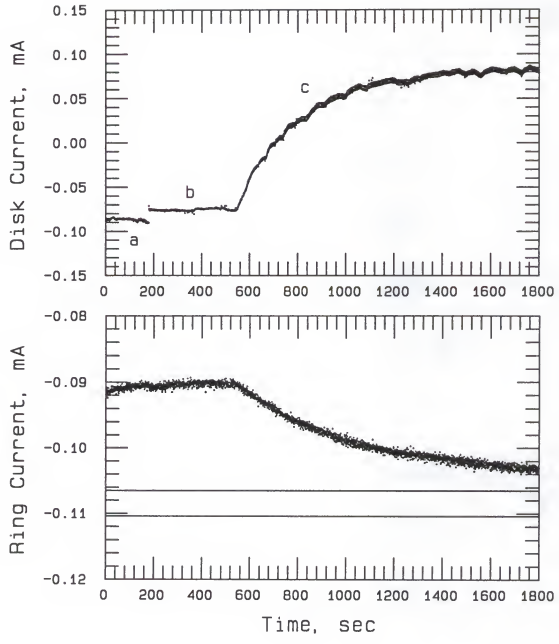


Figure 3.23: Disk and ring currents for the potential step experiment (−0.35V (Ag/AgCl)) using the same experimental system described in Figure 3.21. a) I_d = −0.6V (Ag/AgCl) without illumination. b) I_d = −0.35V (Ag/AgCl) without illumination. c) I_d = −0.35V (Ag/AgCl) with illumination.

(Ag/AgCl) and dark condition. The values of the shielding factor ranged from 1.00 to 1.07 as compared to the theoretical value of 1.10. The solid lines drawn across the ring current transient correspond to the calculated unshielded ring currents using equation 3.36. The unshielded ring currents represented by the *upper* lines are based on the experimental collection efficiency $N_o = 0.18$, as measured from previous experiments. The unshielded ring currents represented by the *lower* lines are based on the theoretical value of collection efficiency, $N_o = 0.225$. Leakage currents were encountered with the Pt-ring electrode (without a disk present) which prevented independent measurement of the unshielded ring current. However, the calculated values represent a "window" for determining whether the ring current reverts to an unshielded value after the potential step at dark conditions.

For a potential step to +0.1V, the disk- and ring-current transients illustrated in Figure 3.21 show three distinct regions. In region (a), the shielding factor was estimated to be 1.07 and suggests that the disk electrode shielded the ring electrode by reducing ferric ions. In region (b), upon the potential step to +0.1V in the dark, the ring-current transient reverts to the unshielded value while the disk current was essentially zero. Under illumination in region (c), the GaAs disk attained a constant photocurrent value. The ring transient exceeded the unshielded ring current value and increased almost linearly with time. This meant that ferrous ions were being oxidized at the GaAs disk electrode and was competing with the photodissolution process of the semiconductor.

Mass-balance calculations have shown that the transient behavior of the ring current could not be attributed to an increase in the bulk ferric concentration due to the "uncollected" ferric ions reaching the ring electrode. The change in ferric concentration in the bulk is given by

$$nFV \frac{d\,C_{Fe^{+3}}}{dt} = (1 - N_o) I_d \tag{3.42}$$

where V is the volume of the solution, $n = 1$ equiv/mol, F is Faraday's constant, and $(1 - N_o)I_d$ is proportional to the "uncollected" ferric ions produced at the disk. Using Levich's equation for a ring electrode, it can be shown that

$$\frac{d\,C_{Fe^{+3}}}{dt} = \frac{1}{K_r}\frac{d\,I_r}{dt} \tag{3.43}$$

where $K_r = 0.62nF\pi(r_3^3 - r_2^3)^{2/3}D_i^{2/3}\nu^{-1/6}\Omega^{1/2}$. Therefore, the disk current required to effect a change in the ring measurements is given by

$$I_d = \frac{nFV}{(1 - N_o)}\frac{1}{K_r}\frac{d\,I_r}{dt}. \tag{3.44}$$

In region (c) of Figure 3.21, the change in ring current over time was found to be $dI_r/dt = 5.7 \times 10^{-9}$ A/s. The density of the electrolyte was 1.105 gm/cm^3 and the volume V of the beaker was 210 cm^3. The collection efficiency of the RRDE was set to $N_o = 0.182$. Assuming that the viscosity of the solution is that of water at 25°C, and that $D_i = 0.5 \times 10^{-5}$ cm^2/s, the disk current I_d was estimated to be 2.0 mA. This value represents the disk current required to accumulate ferric ions in the bulk in order to account for the change in the ring transient. This is an order of magnitude greater than the measured photocurrent at the GaAs disk electrode. Consequently, it was concluded the ring transient was characteristic of a time-dependent stabilization process at the GaAs disk.

This time dependent behavior is illustrated in Figure 3.24 for the deconvoluted redox and GaAs decomposition reactions at the disk electrode. The partial anodic currents were estimated using a collection efficiency of 0.20. The error bars in each graph represent the uncertainty if the experimental value of N_o were between 0.18 and 0.225. Although the total disk current was nearly constant, these graphs show that the rate of the dissolution reaction decreased with time while the rate of the stabilization reaction increased.

The disk- and ring-current transients for a potential step to −0.25V (Ag/AgCl) is illustrated in Figure 3.22. At this potential, the ring current does not revert to the

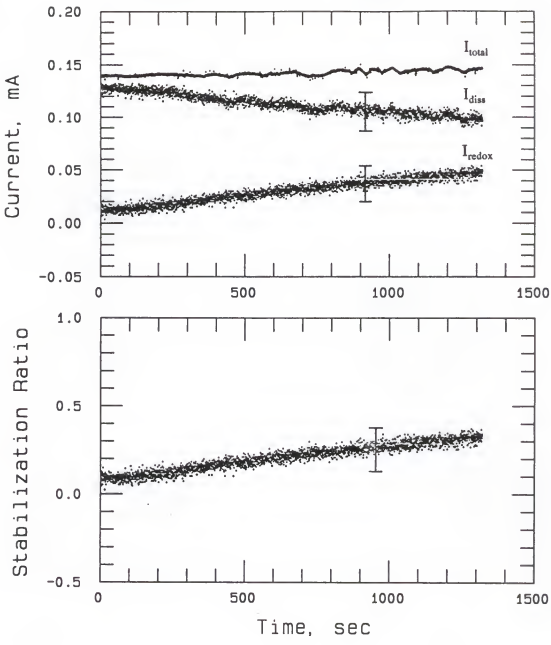


Figure 3.24: Partial disk currents and stabilization ratio for the experimental system described in Figure 3.21 when the disk potential is +0.1V (Ag/AgCl). a) Deconvoluted redox and GaAs decomposition currents. b) Calculated stabilization ratio, assuming $N_o = 0.20$.

unshielded value. This behavior indicated that stabilization is not occurring at the GaAs disk electrode.

For a specified potential of -0.35V, the current response of the GaAs disk to the potential perturbation was clearly time dependent as illustrated in Figure 3.23. Under dark conditions, the potential step did not influence the rate of ferric reduction at the disk as shown by a constant ring current in regions (a) and (b), although a more positive disk current indicated that corrosion was also taking place. Upon illumination, the disk current took a relatively long time to reach a limiting value, with a concurrent unshielding of the ring electrode. Referring to the potentiodynamic scans from the previous section, a disk potential of -0.35V is below the light-limited current plateau. The rate of electron–hole pair recombination becomes important due to a reduced potential difference within the space charge region. At this cathodic potential, the recombination processes at the surface seem to have limited the rate of the corrosion reactions initially. The reaction mechanism of Vanmaelkelbergh *et al.* [85] suggested that corrosion intermediates participate in the decomposition process. Consequently, as corrosion intermediates or products formed on the clean pre-etched GaAs surface, the rate of photocorrosion increased correspondingly. This time-dependent behavior is presented in Figure 3.25 as partial reactions at the disk electrode.

Additional potentiostatic experiments were performed, and the current transients for the GaAs-disk and Pt-ring electrodes at specified potentials are compiled in Appendix D. The specified potentials were $+0.1$V, 0.0V, -0.1V, -0.2V, -0.25V, -0.3V, and -0.35V (Ag/AgCl), and the potentiostatic data are presented in Figures D.1 to D.4, respectively. The deconvoluted partial disk reactions for potentials which exhibited stabilization under illumination are illustrated in Figures D.8 to D.11, and in Figure D.12 for the potential step to -0.35V.

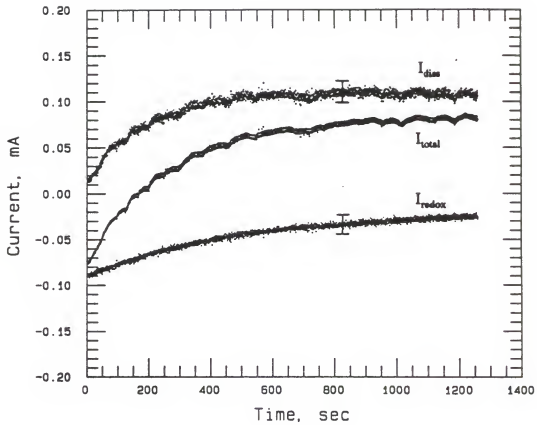


Figure 3.25: Partial disk currents for the experimental system described in Figure 3.21 when the disk potential is -0.35V (Ag/AgCl).

CHAPTER 4
CONCLUSIONS

The scope of this dissertation was the development of an algorithm based on the Kramers–Kronig relations to test the consistency of electrochemical impedance spectra and the investigation of photoelectrochemical etching of n-GaAs. The conclusions of these studies are presented separately.

4.1 Application of the KK–Relations

The work on the Kramers–Kronig relations proposed a solution to the controversy concerning the use of these equations to impedance data that do not exhibit a DC limit. The mathematical development of the algorithm follows from the idea that, through the KK–equations, a unique transformation exists if one component is known over all frequencies. Consequently, if data were available for both components within a finite frequency domain, then it should be possible to calculate the unmeasured impedance from these same equations by assuming an approximating function applicable within the inaccessible frequency domain. This concept has been implemented using the equations presented in Appendix A, and its application to impedance data, both synthetic and experimental, has been illustrated in Chapter 2. The conclusions of the study are given below.

- *The technique was shown to be sensitive to testing violations of the stability criterion.* It is particularly useful to the experimenter to know whether data are

characteristic of a linear and stable system. The examples presented in Chapter 2 showed that the technique is sensitive to testing violations of the stability condition. The appearance of a discontinuity between the calculated impedance values and the experimental data at ω_{min} was shown to be characteristic of inconsistent data which may have violated one or more of the conditions implicit in the KK–relations.

- *The algorithm can identify a cut-off frequency for data sets of pseudo-stationary systems.* For impedance data which are known to be characteristic of non–stationary systems, the KK–algorithm can identify a cut-off frequency above which the data are known to satisfy the KK-relations. Analysis of data in this manner provide the experimenter with "snapshots" of impedance data appropriate for a pseudo-stationary system. This approach was illustrated for the metal hydride system and rotating copper disk electrode.

- *This method complements other validation techniques such equivalent circuit fitting.* The algorithm provides a convenient technique for validating experimental data before performing an equivalent circuit fit. The approach presented here complements this method of analysis because it can identify whether the data set is suitable for treatment by a fitting procedure to an equivalent electrical circuit. Furthermore, a cut-off frequency can be determined which limits the experimental frequency to a domain where the data are consistent with the KK–relations. Finally, for data sets that do not exhibit a DC limit but extend to a low enough frequency where a maximum in the imaginary impedance is observed, the algorithm yields a good estimate of the low–frequency asymptote that would be comparable to the extrapolation using the fitting parameters of an equivalent circuit. These considerations should facilitate the interpretation and analysis of experimental impedance data.

4.2 Photoelectrochemical Etching of GaAs

The rotating ring–disk electrode is an effective tool for studying the photoelectrochemical etching of semiconductors. This study focused on the current-potential characteristics of an n-GaAs disk Pt-ring electrode immersed in solution containing redox species. The photoelectrochemical behavior of the semiconductor was investigated for two cases. Hole-injection experiments were designed to enhance the etching process by addition of an oxidizing agent. Stabilization experiments studied the role of a reducing agent, such as Fe^{+2}, in the suppression of photodecomposition reactions.

- *Fabrication technique of the RRDE provided reliable current–potential measurements.* The most basic requirement in the use of the RRDE as an electroanalytical tool is to a fabricate an electrode that is leakage–free. Following experimental procedures described by Notten [67], the results of the hole injection experiments assured that the current–potential characteristics of the GaAs-disk Pt-ring electrode were comparable to that reported in the literature.

- *The design of potentiostatic experiments was a useful tool to investigate time–dependent dissolution of GaAs.* The potentiostatic runs were designed to be partially "self-calibrating," but still required values for the collection efficiency of the RRDE which could be obtained through independent measurement of the unshielded ring current. Due to the predictable response of a well fabricated RRDE, qualitative analysis of the data was possible even without accurate values for N_o or $I^o_{r,lim}$.

- *Stabilization at anodic potentials was time–dependent.* The hysteresis observed during potentiodynamic experiments indicate time-dependent processes occurring at the interface which could not be replicated in the forward and reverse directions at moderate sweeprates. Steady–state behavior was not reached for

some potentiostatic experiments even after 20 minutes. Therefore, the long-term behavior of the GaAs disk electrode cannot be ascertained by potentiodynamic experiments. The potential-step experiments sought to identify whether the reducing agent Fe^{+2} could stabilize the corrosion of GaAs at a given light intensity and applied potential. It was shown that ring current transients could not be attributed to changes in bulk concentration, but are due instead to changes in the activity of the GaAs disk electrode. Since the total photocurrent was constant under illumination, the stabilization process had to be time-dependent.

- *Surface recombination below the light-limited current plateau influenced the corrosion process.* At cathodic potentials, below the light-limited value for photocurrent, the decomposition reactions appeared to be limited by the recombination processes at the semiconductor surface.

- *Electrochemical impedance spectroscopy could be employed to explain the time-dependent behavior.* EIS can be used to identify the various mechanisms which may have different time constants. Although this system is inherently nonstationary, analysis of impedance data using the algorithm described above could determine the validity of the data to represent a valid "snapshot" of a pseudo-stationary system.

Appendix A
THE KRAMERS–KRONIG ALGORITHM

The mathematical development of an algorithm for employing the KK–relations to evaluate the consistency of electrochemical impedance data is presented in this Appendix.

The two forms of the KK–relations utilized in this algorithm are

$$Z_i(\omega) = -\left(\frac{2\omega}{\pi}\right) \int_0^\infty \frac{Z_r(x) - Z_r(\omega)}{x^2 - w^2}\, dx \tag{A.1}$$

$$Z_r(\omega) = Z_r(\infty) + \frac{2}{\pi} \int_0^\infty \frac{x\, Z_i(x) - \omega\, Z_i(\omega)}{x^2 - w^2}\, dx\,. \tag{A.2}$$

$Z_r(\omega)$ and $Z_i(\omega)$ are the real and imaginary components of the impedance, and ω and x are angular frequencies. Equation (A.1) represents the imaginary impedance that is internally consistent with the real component of the experimental data. Equation (A.2) represents the real impedance that is internally consistent with the imaginary component of the experimental data.

A.1 Mathematical Derivation

Based on the assumptions presented in Chapter 2, the infinite integration region for the KK–equations is limited to $\omega_0 \leq x \leq \omega_{max}$. Experimental data are assumed to be available for the frequency range $\omega_{min} \leq x \leq \omega_{max}$; the real component reaches an asymptote $Z_{r,\infty}$ and the imaginary component approaches zero at ω_{max}. The object of the algorithm is to calculate the unmeasured impedance below

ω_{min} using the KK–relations given by equations (A.1) and (A.2). This requires polynomial expressions for the real and imaginary impedance applicable below the lowest measured frequency ω_{min} down to a low-frequency limit ω_0. The approximating functions in the low frequency domain $\omega_0 \leq \omega_l < \omega_{min}$ are given by

$$Z_r(\omega_l) = \sum_{k=0}^{K} a_k^{(1)} (\log \omega_l)^k \tag{A.3}$$

$$Z_i(\omega_l) = \sum_{m=0}^{M} a_m^{(2)} (\log \omega_l)^m \tag{A.4}$$

which, when appended to the experimental data set, force the data to satisfy the KK–equations. These functions are polynomial expressions of order K and M, with coefficients $a_0^{(1)}$ to $a_K^{(1)}$ and $a_0^{(2)}$ to $a_M^{(2)}$, respectively. These functions are also used for piece-wise interpolation of the experimental data for numerical integration, as discussed below. The semi-logarithmic form was chosen because it is well-behaved in the high frequency regions where high-order polynomials [27] (*i.e.*, $Z(\omega) = \sum_{i=0}^{n} a_i \omega^i$) exhibit oscillations. Although both polynomial expressions are well behaved at lower frequencies, the semi-logarithmic form can be used over the entire frequency domain.

The integrals of equations (A.1) and (A.2) are evaluated over two regions: $\omega_0 \leq x < \omega_{min}$ and $\omega_{min} \leq x \leq \omega_{max}$. For a given frequency ω_l, equations (A.3) and (A.4) were substituted into equations (A.1) and (A.2) to give

$$\begin{aligned} \sum_{m=0}^{M} a_m^{(2)} (\log \omega_l)^m = & -\left(\frac{2\omega_l}{\pi}\right) \int_{\omega_0}^{\omega_{min}} \frac{\sum_{k=0}^{K} a_k^{(1)} (\log x)^k - \sum_{k=0}^{K} a_k^{(1)} (\log \omega_l)^k}{x^2 - w_l^2} \, dx \\ & - \left(\frac{2\omega_l}{\pi}\right) \int_{\omega_{min}}^{\omega_{max}} \frac{Z_r(x) - \sum_{k=0}^{K} a_k^{(1)} (\log \omega_l)^k}{x^2 - w_l^2} \, dx \end{aligned} \tag{A.5}$$

and

$$\begin{aligned} \sum_{k=0}^{K} a_k^{(1)} (\log \omega_l)^k = & \; Z_{r,\infty} + \\ & \left(\frac{2}{\pi}\right) \int_{\omega_0}^{\omega_{min}} \frac{x \sum_{m=0}^{M} a_m^{(2)} (\log x)^m - \omega_l \sum_{m=0}^{M} a_m^{(2)} (\log \omega_l)^m}{x^2 - w_l^2} \, dx \\ & + \left(\frac{2}{\pi}\right) \int_{\omega_{min}}^{\omega_{max}} \frac{x \, Z_i(x) - \omega_l \sum_{m=0}^{M} a_m^{(2)} (\log \omega_l)^m}{x^2 - w_l^2} \, dx \; . \end{aligned} \tag{A.6}$$

$Z_r(x)$ and $Z_i(x)$ in the range $\omega_{min} \leq x \leq \omega_{max}$ represent the experimental impedance data.

Equations (A.5) and (A.6) are rearranged to collect like–terms for the polynomial coefficients $a_k^{(1)}$ and $a_m^{(2)}$ such that

$$\sum_{m=0}^{M} a_m^{(2)} (\log \omega_l)^m = -\left(\frac{2\omega_l}{\pi}\right) \int_{\omega_{min}}^{\omega_{max}} \frac{Z_r(x)}{x^2 - w_l^2}\, dx \tag{A.7}$$
$$-\left(\frac{2\omega_l}{\pi}\right) \sum_{k=0}^{K} a_k^{(1)} \left\{ \int_{\omega_0}^{\omega_{min}} \frac{(\log x)^k - (\log \omega_l)^k}{x^2 - w_l^2}\, dx - (\log \omega_l)^k \int_{\omega_{min}}^{\omega_{max}} \frac{dx}{x^2 - w_l^2} \right\},$$

and

$$\sum_{k=0}^{K} a_k^{(1)} (\log \omega_l)^k = Z_{r,\infty} + \left(\frac{2}{\pi}\right) \int_{\omega_{min}}^{\omega_{max}} \frac{x\, Z_i(x)}{x^2 - w_l^2}\, dx \tag{A.8}$$
$$+\left(\frac{2}{\pi}\right) \sum_{m=0}^{M} a_m^{(2)} \left\{ \int_{\omega_0}^{\omega_{min}} \frac{x (\log x)^m - \omega_l (\log \omega_l)^m}{x^2 - w_l^2}\, dx - \omega_l (\log \omega_l)^m \int_{\omega_{min}}^{\omega_{max}} \frac{dx}{x^2 - w_l^2} \right\}.$$

Also,

$$\int_{\omega_{min}}^{\omega_{max}} \frac{dx}{x^2 - w_l^2} = \frac{1}{2\omega_l} \ln\left(\frac{\omega_{max} - \omega_l}{\omega_{max} + \omega_l} \cdot \frac{\omega_{min} + \omega_l}{\omega_{min} - \omega_l}\right). \tag{A.9}$$

Since $\omega_{max} \gg \omega_l$, equation (A.9) can be simplified to

$$\int_{\omega_{min}}^{\omega_{max}} \frac{dx}{x^2 - w_l^2} \simeq \frac{1}{2\omega_l} \ln\left(\frac{\omega_{min} + \omega_l}{\omega_{min} - \omega_l}\right). \tag{A.10}$$

After substituting equation (A.10) into equations (A.7) and (A.8), the following expressions are obtained:

$$a_0^{(1)} C_0^{(1)} + a_1^{(1)} C_1^{(1)} + a_2^{(1)} C_2^{(1)} + \cdots + a_K^{(1)} C_K^{(1)} +$$
$$a_0^{(2)} + a_1^{(2)} \log \omega_l + a_2^{(2)} (\log \omega_l)^2 + \cdots + a_M^{(2)} (\log \omega_l)^M = \tag{A.11}$$
$$-\left(\frac{2\omega_l}{\pi}\right) \int_{\omega_{min}}^{\omega_{max}} \frac{Z_r(x)}{x^2 - w_l^2}\, dx$$

where

$$C_k^{(1)}\Big|_{0 \leq k \leq K} = \left(\frac{2\omega_l}{\pi}\right) \left[\int_{\omega_0}^{\omega_{min}} \frac{(\log x)^k - (\log \omega_l)^k}{x^2 - w_l^2}\, dx - \frac{(\log \omega_l)^k}{2\omega_l} \ln\left(\frac{\omega_{min} + \omega_l}{\omega_{min} - \omega_l}\right) \right]$$

and

$$a_0^{(1)} + a_1^{(1)} \log \omega_l + a_2^{(1)} (\log \omega_l)^2 + \cdots + a_K^{(1)} (\log \omega_l)^K + a_0^{(2)} C_0^{(2)} + a_1^{(2)} C_1^{(2)} + a_2^{(2)} C_2^{(2)} + \cdots + a_M^{(2)} C_M^{(2)} = Z_r(\infty) + \left(\frac{2}{\pi}\right) \int_{\omega_{min}}^{\omega_{max}} \frac{x\, Z_i(x)}{x^2 - w_l^2}\, dx \qquad \text{(A.12)}$$

where

$$C_m^{(2)}\Big|_{0 \le m \le M} = -\left(\frac{2}{\pi}\right) \left[\int_{\omega_0}^{\omega_{min}} \frac{x (\log x)^m - \omega_l (\log \omega_l)^m}{x^2 - w_l^2}\, dx - \frac{(\log \omega_l)^m}{2} \ln\left(\frac{\omega_{min} + \omega_l}{\omega_{min} - \omega_l}\right) \right].$$

Equations (A.1) and (A.2) have been reduced to algebraic equations (A.11) and (A.12), respectively, where the polynomial coefficients $a_0^{(1)}$ to $a_K^{(1)}$ and $a_0^{(2)}$ to $a_M^{(2)}$ comprise the $(K + M + 2)$ unknowns to be determined. Therefore, $(K + M + 2)$ independent equations must be obtained at frequencies within the interval $\omega_0 \le \omega_l < \omega_{min}$, where $1 \le l \le (K + M + 2)$. This is conveniently achieved by evenly dividing $\omega_0 \le x < \omega_{min}$ in logarithmic scale and assigning the $(K + M + 2)$ values of frequency to ω_l.

The integrals of equations (A.11) and (A.12) can be evaluated using Simpson's Rule by following the procedure suggested by Urquidi-Macdonald *et al.* [27] or by an adaptive quadrature algorithm. These techniques are discussed in the next section. It can also be shown, using l'Hospital's rule, that the transform integrands involved in calculating C_k and C_m have finite, non-zero values at $x = w_l$ [39]. Piecewise polynomial functions, as in equations (A.3) and (A.4), are also required to evaluate the integrals involving the discrete experimental impedance data in the frequency domain $\omega_{min} \le x \le \omega_{max}$. These functions can be determined by least-squares regression.

Equations (A.11) and (A.12) are algebraically coupled by the interdependence of the impedance components. This means that the polynomial coefficients calculated from each equation must be equal. The two equations could be solved independently and must yield equivalent values for the polynomial coefficients. However, discrepancies may arise due to round-off errors inherent in the least-squares regression of the experimental data, in the numerical integration procedures, and in calculating the coefficients $C_k^{(1)}$ and $C_m^{(2)}$. An iterative refinement algorithm must be implemented to match the set of polynomial coefficients for both equations (for example, see Algorithm 7.4 [123]).

This iterative routine may be avoided by using equations (A.11) and (A.12) simultaneously when constructing the linear system of $(K + M + 2)$ equations. For example, equation (A.11) is employed when l is odd, and equation (A.12) when l is even. This direct method yields polynomial coefficients that are valid for both equations and which result in residuals of comparable orders of magnitude. This was the approach taken in the development of the computer software program.

A.2 Numerical Integration of Integrands

This section presents the equations for the numerical integration of the KK–relations, namely, the trapezoidal method and Gaussian Quadrature.

A.2.1 Limits of Integrands at $x = \omega$

The integrands of the KK-relations used for the algorithm presented here do not contain singularities at $x = \omega_l$ contrary to the statement of Urquidi-Macdonald *et al.* [27, 28]. Using l'Hospital's rule, Townley [39] showed that

$$\lim_{x \to \omega} \frac{Z_r(x) - Z_r(\omega)}{x^2 - w^2} = \frac{1}{2\omega} \cdot \frac{dZ_r(\omega)}{d\omega} \tag{A.13}$$

and

$$\lim_{x \to \omega} \frac{x Z_i(x) - \omega Z_i(\omega)}{x^2 - w^2} = \frac{1}{2\omega} \cdot \frac{d[\omega Z_i(\omega)]}{d\omega} . \tag{A.14}$$

Since the impedance spectrum is continuous in the frequency domain, the integrands of equations (A.1) and (A.2) have finite, nonzero values at $x = w$. It can be shown that

$$\lim_{x \to \omega_l} \frac{(\log x)^k - (\log \omega_l)^k}{x^2 - \omega_l^2} = \frac{k (\log \omega_l)^k}{2\omega_l^2 \ln \omega_l} \tag{A.15}$$

and

$$\lim_{x \to \omega_l} \frac{x(\log x)^m - \omega_l (\log \omega_l)^m}{x^2 - \omega_l^2} = \frac{(\log \omega_l)^m}{2\omega_l} \cdot \left(1 + \frac{m}{\ln \omega_l}\right) . \tag{A.16}$$

Equations (A.15) and (A.16) are used to evaluate the integrands for the coefficients $C_k^{(1)}$ and $C_m^{(2)}$ of equations (A.11) and (A.12), respectively, when $x = w_l$.

A.2.2 Trapezoidal Method: Simpson's Rule

The technique suggested by Urquidi-Macdonald *et al.* [27] for evaluating the integrals was employed in an earlier version of the computer software (see Esteban and Orazem [50] and Orazem *et al.* [51]). This involves subdividing the regions $\omega_0 \leq x < \omega_{min}$ and $\omega_{min} \leq x \leq \omega_{max}$ into several segments of evenly spaced frequencies as illustrated in Figure 2.1. N_c represents the number of segments in the low–frequency range, and N_e the number of segments in the experimental frequency range. Within each segment in the frequency range $x_{min} \leq x \leq x_{max}$, the interval is uniformly spaced over δ_i divisions such that

$$\Delta X_i = \frac{x_{max} - x_{min}}{\delta_i} . \tag{A.17}$$

Polynomial expressions, as in equations (A.3) and (A.4), were obtained for the discrete experimental impedance data using least–squares approximation. These equations were used to evaluate $Z_r(x)$ and $Z_i(x)$ within the experimental data segments at

uniformly spaced frequencies. The integrals of equation (A.11) were approximated by discrete summations within each segment to obtain

$$\int_{\omega_{min}}^{\omega_{max}} \frac{Z_r(x)}{x^2 - w_l^2}\, dx \;=\; \sum_{j=1}^{N_s} \sum_{n=1}^{\delta_j} \frac{Z_r(x_n)}{x_n^2 - w_l^2}\, \Delta X_j \tag{A.18}$$

and

$$\int_{\omega_0}^{\omega_{min}} \frac{(\log x)^k - (\log \omega_l)^k}{x^2 - w_l^2}\, dx \;=\; \sum_{j=1}^{N_s} \sum_{n=1}^{\delta_j} \frac{Z_r(x_n)}{x_n^2 - w_l^2}\, \Delta X_j\,. \tag{A.19}$$

Analogous expressions for equation (A.12) can be derived such that

$$\int_{\omega_{min}}^{\omega_{max}} \frac{x\, Z_i(x)}{x^2 - w_l^2}\, dx \;=\; \sum_{j=1}^{N_s} \sum_{n=1}^{\delta_j} \frac{x_n\, Z_i(x_n)}{x_n^2 - w_l^2}\, \Delta X_j \tag{A.20}$$

and

$$\int_{\omega_0}^{\omega_{min}} \frac{x\,(\log x)^m - \omega_l\,(\log \omega_l)^m}{x^2 - w_l^2}\, dx \;=\; \sum_{i=1}^{N_c} \sum_{n=1}^{\delta_i} \frac{x_n\,(\log x_n)^m - \omega_l\,(\log \omega_l)^m}{x_n^2 - w_l^2}\, \Delta X_i \tag{A.21}$$

A.2.3 Gaussian Quadrature

The computer program documented in Appendix B employs an adaptive fifth-order Gaussian Quadrature to evaluate the integrals of the KK-algorithm. The relation

$$\int_a^b f(x)dx \;=\; \frac{b-a}{2} \int_{-1}^{1} f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right) dt \tag{A.22}$$

uses the linear transformation $t = (2x - a - b)/(b - a)$ to translate any interval $[a, b]$ into $[-1, 1]$. The Legendre polynomials can then be used to approximate the integral such that

$$\int_{-1}^{1} P(x)dx \;=\; \sum_{i=1}^{n} c_i P(x_i) \tag{A.23}$$

where

$$c_i \;=\; \int_{-1}^{1} \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)} dx\,, \tag{A.24}$$

$x_0, x_1, \ldots, x_n$ are the zeroes (or roots) of the nth order Legendre polynomial, and P is any polynomial of degree less than or equal to $2n-1$[123]. Tabulated values for the coefficients c_i and the roots of the Legendre polynomials are found in the literature [123, 124].

A.3 Theoretical Impedance Expressions

The analytic expressions employed for calculating the impedance response of the equivalent circuits depicted in Figure 2.2 are listed in this section.

A.3.1 Circuit 2.2a

The real and imaginary impedance components for the Randles circuit of Figure 2a were calculated from

$$Z_T(\omega) = Z_r(\omega) + j\, Z_i(\omega) = \left(R_s + \frac{R_1}{1+\omega^2 R_1^2 C_1^2} \right) - j\, \frac{\omega R_1^2 C_1}{1+\omega^2 R_1^2 C_1^2}. \quad \text{(A.25)}$$

A.3.2 Circuit 2.2b

The impedance response for the Randles circuit with a Warburg impedance shown in Figure 2b is deconvoluted from the impedance response of the electrical circuit of Figure 2d by reducing the value of the time constant R_2C_2. Consequently, the impedance data are calculated from equation (A.30) presented in Appendix B.4 by setting the values of R_2 and C_2 equal to zero.

A.3.3 Circuit 2.2c

The electrical circuit of Figure 2c was used to model the impedance behavior of mild steel in concentrated hydrochloric acid [125]. It has a constant phase element

(CPE), whose impedance is given by

$$Z_{CPE} = \frac{1}{Y_o(j\omega)^n} \tag{A.26}$$

The rationale behind the use of this element is discussed in References [2] and [15]. Since

$$j^n = \cos(\frac{n\pi}{2}) + j\,\sin(\frac{n\pi}{2}), \tag{A.27}$$

the analytic expression for this equivalent circuit is

$$Z_T(\omega) = Z_r(\omega) + j\,Z_i(\omega) = \left(R_s + \frac{\alpha\,R_1}{\alpha^2+\beta^2}\right) - j\,\frac{\beta\,R_1}{\alpha^2+\beta^2} \tag{A.28}$$

where

$$\alpha = 1 + Y_o R_1 \omega^n \cos(\frac{n\pi}{2})$$
$$\beta = Y_o R_1 \omega^n \sin(\frac{n\pi}{2}).$$

A.3.4 Circuit 2.2d

The equivalent circuit of Figure2.2d was employed to model the impedance response of copper dissolution in alkaline chloride solutions [53]. The expression employed for the Warburg impedance Z_w is given by [16]

$$Z_w = R_w \frac{\tanh(\sqrt{j\omega T_w})}{\sqrt{j\omega T_w}}. \tag{A.29}$$

The analytic impedance response of the equivalent circuit shown in Figure 2d is

$$Z_T(\omega) = Z_r(\omega) + j\,Z_i(\omega) = \left(R_s + \frac{\Lambda}{\Delta^2+\Sigma^2}\right) - j\,\frac{\Omega}{\Delta^2+\Sigma^2} \tag{A.30}$$

where

$$\Omega = \rho\Sigma - \psi\Delta$$
$$\Lambda = \rho\Delta - \psi\Sigma$$

$$
\begin{aligned}
\Sigma &= \omega C_1 \rho \\
\Delta &= \kappa - \omega C_1 \psi \\
\psi &= R_w \eta \mu - \phi \omega R_2 \tau_2 \zeta \\
\rho &= R_1 \kappa + R_w \lambda \mu + \phi \zeta R_2 \\
\kappa &= \phi \zeta \mu \\
\mu &= 1 + \omega^2 \tau_2^2 \\
\zeta &= \delta^2 + \sigma^2 \\
\tau_2 &= R_2 C_2 \\
\eta &= \gamma \delta \sin\phi - \beta \sigma \cos\phi \\
\lambda &= \beta \delta \cos\phi + \gamma \sigma \sin\phi \\
\sigma &= \beta \sin\phi + \gamma \cos\phi \\
\delta &= \gamma \cos\phi - \beta \sin\phi \\
\gamma &= e^{\phi} + e^{-\phi} \\
\beta &= e^{\phi} - e^{-\phi} \\
\phi &= \frac{\sqrt{2\omega T_w}}{2} .
\end{aligned}
$$

A.3.5 Simulation of inconsistent impedance data

Inconsistent impedance data for a non-stationary system were simulated by changing the value of a circuit element during the course of the "experiment". A simple case is to assume that the value of the circuit element E varies proportionally with time. Therefore,

$$E_i = E_o + f N \sum_{n=1}^{i} \Delta t_n , \qquad \text{(A.31)}$$

where E_o is the value of the circuit element at start of the experiment, i is the number of data points collected, and E_i is the value of E as the experiment progresses. The

"inconsistency factor" f accounts for the direction of change of E; the value of E decreases with time when $f < 0$, and increases when $f > 0$. N represents the number of cycles used to make measurement at a given frequency. The time required to complete one cycle is $\Delta t = 1/\omega$, and $N\Delta t = N/\omega$ corresponds to the total time which has elapsed when the measurement was performed. The expression for E_i becomes

$$E_i = E_o + fN \sum_{n=1}^{i} \frac{1}{\omega_n}, \tag{A.32}$$

where $w_1 = 65$ kHz since impedance experiments are usually performed from 65 kHz down to low frequency values. The percent difference between the initial value of the circuit element and its value at any time during the "experiment" is given by

$$\Delta E_{\%} = \frac{E_i - E_o}{E_o} \times 100 = \frac{fN}{E_o} \sum_{n=1}^{i} \frac{1}{\omega_n} \times 100. \tag{A.33}$$

Equation (A.32) was used to calculate the values of a variable resistor or variable capacitor, and was incorporated into the analytic expressions listed above to simulate a non–stationary system. The value of N was set to 6 when generating the inconsistent data sets.

Appendix B
KRAMERS–KRONIG PROGRAM

B.1 Main Program

```
program Main_Program (input, output);
(*******************************************************

     CALCULATING LOW FREQUENCY IMPEDANCE USING THE
           KRAMERS-KRONIG (BODE) RELATIONS

        J. Matthew R. Esteban and Mark E. Orazem
           Department of Chemical Engineering
                 University of Florida
                 Gainesville, FL 32611
                   (C) COPYRIGHT 1990

*******************************************************)

{$N+}          { 8087 Compiled }
{$X+}          { extended function calls }

uses
   Dos, Graph, CRT, Printer,
   KKDecl, InputKK, GrafKK, TwoGraf, Gauss,
   Files, KKEqns, GetSlope, DataFit;
var
   ReturnMain: boolean;

procedure Calculate_Low_Frequency_Z;
var
   Delta: extended;
   i, j: integer;
   Abort: boolean;
begin
   Abort := FALSE;
   RetrieveDataFile (Abort);
   if NOT Abort then begin
```

```
      InputParameters (Abort);
      if NOT Abort then
         Perform_Regression_and_Restructure_Frequency_Arrays (Abort);
      if Not Abort then Solve_Equations_12_and_13;
      end;
end;

procedure WriteBeepToggle;
begin
   if BeepON then
      WriteXY (32,18, 'ON ')
   else
      WriteXY (32,18, 'OFF')
end;

procedure BeepToggle;
begin
   BeepON := NOT BeepON;
   if BeepON then Beeper;
   WriteBeepToggle;
   GotoXY (28,22)
end;

procedure Main_Screen;
begin
   TextAttr := Yellow + Blue*16;   ClrScr;
   Center (2, 'CALCULATING LOW FREQUENCY IMPEDANCE DATA');
   Center (3, '----------------------------------------' +
              '------------------------');
   Center (4, 'Test internal consistency of data using ' +
              'Kramers-Kronig relations');
   WriteXY (13, 6, '1 )  Calculate Low Frequency Impedance Data');
   WriteXY (13, 8, '2 )  Replot Results');
   WriteXY (13,10, '3 )  Calculate slope at truncation frequency');
   WriteXY (13,12, 'R )  Read (R) impedance data from disk');
   WriteXY (13,14, 'S )  Store New Results');
   WriteXY (13,16, 'D )  Delete Points');
   WriteXY (13,18, 'B )  Toggle BEEP:  ');
   WriteBeepToggle;
   WriteXY (13,20, 'E )  Terminate program ...EXIT');
   WriteXY (13,22, 'Enter Choice:  ');
end;  { Procedure Read_KBD_screen }

procedure Initialization;
var
   i: integer;
```

```
begin
   HaveData  := FALSE;
   BeepOn    := FALSE;    { Beep turned off }
   AutoPlot  := TRUE;
   LogXaxis  := TRUE;
   LogYaxis  := FALSE;
   TestMode  := ' ';
   FileName  := '';
   GetDir (0, DirPath);   { Get current Directory }
   AllocateDynamicArrays;
   for i := 1 to MaxSegments do  begin
       Index[i] := 0;
       Low_AB[i] := 0.0
       end;
   OrigVideoMode := LastMode;   { Store Original video mode }
   Tg[1]:= 0.9061798459;  Ag[1] := 0.2369268850;
   Tg[2]:= 0.5384693101;  Ag[2] := 0.4786286705;
   Tg[3]:= 0.0;           Ag[3] := 0.5688888889;
   Tg[4]:= -Tg[2];        Ag[4] := Ag[2];
   Tg[5]:= -Tg[1];        Ag[5] := Ag[1];
   InitializeGraphics;
end;

procedure NoData;
begin
   if BeepOn then Beeper;
   WriteXY (13,22,'NO DATA AVAILABLE ... Retrieve Data File !!!');
   delay (1500)
end;

begin
   Initialization;
   repeat
      ReturnMain := FALSE;
      if TestMode <> 'B' then Main_Screen;
      Read_KBD (TestMode);
      case TestMode of
        '1':  Calculate_Low_Frequency_Z;
        '2':  if HaveData then Plot_Restored_Impedances else NoData;
        '3':  if HaveData then Calculate_Slopes else NoData;
        'R':  Retrieve_data (ReturnMain);
        'S':  if HaveData then Store_Results else NoData;
        'D':  if HaveData then DeletePoints else NoData;
        'B':  BeepToggle;
        'E':  writeln ('< PROGRAM HAS BEEN TERMINATED >');
      end;   { CASE }
```

```
   until TestMode = 'E';
   GoodBye
end.      { Main Program }
```

B.2 Variable Declarations

```
unit KKDecl;
(*********************************************
     Variable and function declarations
     for Kramers-Kronig program
*********************************************)

{$N+,E+,X+}

interface

uses Crt, Graph;

const  CorrCoefOK            =  0.99985;
       MaxDataPts            =  200;
       Max_Poly_order        =  30;
       Twice_Max_poly_order  =  2*Max_Poly_Order;
       MaxSegments           =  15;
       MaxSegmentDivs        =  160;
       Store                 =  1;
       Retrieve              =  2;
       RealData              =  1;
       ImagData              =  2;
       Calculated            =  1;
       Experimental          =  2;
       Normal                =  1;
       Restore               =  2;
       CR                    =  1;
       ESC                   =  2;
       HOME                  =  3;
       ENDKey                =  4;
       UP                    =  5;
       LEFT                  =  6;
       RIGHT                 =  7;
       DOWN                  =  8;
       Arrow: string[4]      =  '<---';
       GAUSS_ORDER           =  5;

type   Vector         =  array [0..Max_Poly_Order] of extended;
       Sum_X_vector   =  array [1..Twice_Max_Poly_Order] of extended;
```

```
        NrowVector      =  array [0..Max_Poly_Order] of integer;
        IntArray        =  array [0..MaxSegments] of integer;
        ExtArray        =  array [0..MaxSegments] of extended;
        MatrixA         =  array [0..Max_Poly_order, 0..Max_Poly_Order]
                           of extended;
        Data_Array      =  array [-Max_Poly_Order..MaxDataPts]
                           of extended;
        GaussQuadArray  =  array [1..GAUSS_ORDER] of extended;
        Frequency       =  array [0..MaxSegmentDivs] of extended;
        FreqArrayRec    =  Record
                             w: Frequency;
                           end;
        PolyFitRecord   =  Record
                             Order: integer;
                             r: extended;
                             Coef : Vector
                           end;
        DataRec         =  Record
                             Zr, Zi, w: extended
                           end;
        StorePolyFit    =  array [1..MaxSegments] of PolyFitRecord;
        PolyFitPtr      = ^StorePolyFit;
        SumXnPtr        = ^Sum_X_vector;
        MatPtr          = ^MatrixA;
        GaussVectPtr    = ^Vector;
        DataPtr         = ^Data_Array;
        ExptFreq        =  array [1..MaxSegments] of FreqArrayRec;
        LowFreq         =  array [1..MaxSegments-5] of FreqArrayRec;
        ExptFreqPtr     = ^ExptFreq;
        LowFreqPtr      = ^LowFreq;
        Str20           =  string[20];
        Str40           =  string[40];
        Str80           =  string[80];

var     Poly_coef,
        Zr_Coef, Zi_coef   :  Vector;
        Zr_fit,  Zi_fit    :  PolyFitPtr;
        ARFit              :  PolyFitRecord;  { AutoRegress Best Fit }
        Nrow               :  NrowVector;
        Sum_Xn             :  SumXnPtr;       { Guassian Elimination }
        B, S, Sum_YXn      :  GaussVectPtr;
        A                  :  MatPtr;         { Normal Equation }
        Index              :  IntArray;
        Low_AB             :  ExtArray;       { integration limits }
        Ag, Tg             :  GaussQuadArray; { Gaussian Quadrature }
        Expt_w             :  ExptFreqPtr;
```

```
        Low_w              :  LowFreqPtr;

        OrigVideoMode,
        FitOrder,                              { Gauss. El. order }
        NumPts,                                { # of data points }
        NewPts,                                { # of calc. pts   }
        PolyK, PolyM,                          { poly orders      }
        N1, Ne             :  integer;         { integration segments }

        w, Log_w, w_Hz,                        { Data Arrays }
        Real_Z, Imag_Z     :  DataPtr;
        Zi0, ZrMax         :  DataRec;

        DirPath            :  String;
        FileName           :  Str20;
        choice, TestMode   :  char;

        HaveData, NewData,
        BeepON,
        AutoRegress,
        FastAuto,
        AutoIndex,
        HaveParameters,
        Changed_PolyOrder,
        Changed_w0,
        Changed_SegDivs,
        Inductive,
        InductiveDone      :  Boolean;

        Zr_oo,
        w_0, w_0_Hz,
        Hz_factor, Zfit_Hz :  extended;

function  AntiLOG (X: extended): extended;
function  LOG (X: extended): extended;
function  Poly_Y (Poly_Order: integer; var Poly_Coef: Vector;
                  X: extended): extended;
function  ScanKey: integer;
procedure AllocateDynamicArrays;
procedure AnyKey (Prompt: string);
procedure Beeper;
procedure Blinking_Label (Title: Str80);
procedure Center (Y: integer; StringText: String);
procedure CRKey (Prompt: String);
procedure DeallocateDynamicArrays;
procedure DynamicPointersToNil;
```

```
procedure GetString (var StrInput: Str20; var Abort: Boolean);
procedure GoodBye;
procedure Read_KBD (var  choice: char);
procedure Swap (var a, b: extended);
procedure SwapINT (var a, b: integer);
procedure WriteBlank (X,Y: integer);
procedure WriteXY (X, Y: integer; StrText: string);
procedure YesNo (Prompt: String; var Choice: char);
procedure YNA (Prompt: String; var Choice: char; var Abort: Boolean);

implementation

function ScanKey: integer;
var
   KeyInput: char;
   KeyOK: Boolean;
begin
   repeat
      KeyOK := TRUE;
      repeat
         KeyInput := ReadKey
      until KeyInput in [#0, #13, #27];
      if KeyInput = #0 then
         KeyInput := ReadKey;   { Extended Scan Code }
      case KeyInput of
           #13:  ScanKey := CR;
           #27:  ScanKey := ESC;
           #71:  ScanKey := HOME;
           #79:  ScanKey := ENDKey;
           #72:  ScanKey := UP;
           #75:  ScanKey := LEFT;
           #77:  ScanKey := RIGHT;
           #80:  ScanKey := DOWN;
          else   begin
                    write(^g);
                    KeyOK := FALSE
                 end
      end  { case }
   until KeyOK;
end;

function  LOG (X: extended): extended;
begin
   LOG := LN(X)/LN(10.0);
end;
```

```
function AntiLOG (X: extended): extended;
begin
   AntiLOG := EXP(X*LN(10.0))
end;

function Poly_Y (Poly_Order: integer; var Poly_Coef: Vector;
                 X: extended): extended;
var
   i: integer;
   Sum: extended;
begin
   Sum := 0.;
   for i := 0 to Poly_Order do
       Sum  :=  Sum + Poly_Coef[i]*EXP(i*LN(X));
   Poly_Y :=  Sum
end;

procedure Beeper;
begin
   write (^g)
end;

procedure WriteBlank (X, Y: integer);
begin
   gotoXY (X, Y);
   ClrEOL;
end;

procedure Read_KBD (var  choice: char);
var
   Valid: Boolean;
begin
   repeat
     Valid := TRUE;
     case ReadKey of
       #0  : choice := #0;       { Scan code expected }
       #13 : choice := '^';
       #27 : choice := 'A';      { ABORT }
       #49 : choice := '1';
       #50 : choice := '2';
       #51 : choice := '3';
       #52 : choice := '4';
       #53 : choice := '5';
       #54 : choice := '6';
       #55 : choice := '7';
       #56 : choice := '8';
```

```
        #57 : choice := '9';
        #48 : choice := '0';
        #46 : choice := '.';
        #43 : choice := '+';
        #45 : choice := '-';
        #8  : choice := 'B';        { BackSpace }
        #66, #98  : choice := 'B';  { Beep Toggle }
        #82, #114 : choice := 'R';  { Retrieve file }
        #83, #115 : choice := 'S';  { Store Data }
        #68, #100 : choice := 'D';  { Delete points }
        #69, #101 : choice := 'E';  { Exit or exponent }
        #89, #121 : choice := 'Y';
        #78, #110 : choice := 'N';
      else
        begin
        if BeepOn then Beeper;
        Valid := FALSE
        end
      end;   { CASE }
    until Valid;
end;  { Procedure Read_KBD }

procedure  GetString (var StrInput: Str20; var Abort: boolean);
var
    i, j: integer;
    InChar: char;
    CharArray: array [1..20] of char;
begin
    StrInput := '';
    i := 0;
    repeat
       InChar := ReadKey;
       case Inchar of
          #8:  begin
               CharArray[i] := ' ';
               DEC (i); gotoxy (WhereX-1, WhereY); ClrEOL
               end;
         #13:  { do nothing };
         #27:  Abort := TRUE;
        else   begin
               INC (i); CharArray[i] := Inchar; write (CharArray[i]);
               end
       end; { case }
    until (Inchar = #13) or Abort;
    if Abort
       then StrInput := ''
```

```
      else for j := 1 to i do
              StrInput := StrInput + CharArray[j]
end;

procedure AnyKey (Prompt: String);
var
   dummy: char;
begin
   if Prompt <> '' then begin
     WriteBlank (10,24);
     write (Prompt); writeln
     end;
   repeat until keypressed;
   while keypressed do
      dummy := ReadKey    { empty keyboard buffer }
end;

procedure CRKey (Prompt: String);
begin
   if Prompt <> '' then begin
     WriteBlank (10,24);
     write (Prompt); writeln
     end;
   repeat Read_KBD(Choice) until Choice = '^';
end;

procedure YesNo (Prompt: String; var Choice: char);
begin
   WriteBlank (5,24);
   if BeepON then Beeper;
   write (Prompt);
   repeat Read_KBD(Choice) until Choice in ['Y', 'N', '^'];
   WriteBlank (1,24)
end;

procedure YNA (Prompt: String; var Choice: char; var Abort: Boolean);
begin
   WriteBlank (5,24);
   if BeepON then Beeper;
   write (Prompt);
   repeat Read_KBD(Choice) until Choice in ['Y', 'N', 'A', '^'];
   if Choice = 'A'
      then Abort := TRUE
      else Abort := FALSE;
   WriteBlank (1,24)
end;
```

```
procedure WriteXY (X, Y: integer; StrText: string);
begin
   gotoxy (X,Y); ClrEOL;
   write (StrText)
end;

procedure Center (Y: integer; StringText: String);
begin
   gotoXY (40-(Length(StringText) DIV 2), Y);
   write (StringText)
end;

procedure Blinking_label (Title: Str80);
begin
   TextAttr := Red + Blue*16 + Blink;
   if BeepON then Beeper;
   writeln;
   Center(2, Title);
   TextAttr := Yellow + Blue*16;
end;

procedure Swap (var a, b: extended);
var
   t: extended;
begin
   t := a;
   a := b;
   b := t
end;

procedure SwapINT (var a, b: integer);
var
   c: integer;
begin
   c := a;
   a := b;
   b := c
end;

procedure DynamicPointersToNil;
{ Assign data file pointers to nil }
begin
   w      := nil;
   Log_w  := nil;
   w_Hz   := nil;
```

```
   Real_Z := nil;
   Imag_Z := nil;
   Zr_fit := nil;
   Zi_fit := nil;
end;

procedure AllocateDynamicArrays;
begin
   DynamicPointersToNil;
   new (w);
   new (Log_w);
   new (w_Hz);
   new (Real_Z);
   new (Imag_Z);
   new (Zr_fit);
   new (Zi_fit);
end;

procedure DeallocateDynamicArrays;
begin
   dispose (w);
   dispose (Log_w);
   dispose (w_Hz);
   dispose (Real_Z);
   dispose (Imag_Z);
   dispose (Zr_fit);
   dispose (Zi_fit);
end;

procedure GoodBye;
begin
   DeAllocateDynamicArrays;
   CloseGraph;
   TextMode (OrigVideoMode);
   ClrScr
end;

end.
```

B.3 Equations for KK–Algorithm

```
unit KKEqns;
{
   Subprogram implements the equations for the KK-algorithm.
   The integrals of the KK-relations are evaluated using
```

```
   an adaptive 5th-order Gaussian Quadrature.
}

{$N+}

interface

uses  Crt, Graph, KKDecl, GrafKK, Gauss;

procedure Plot_Restored_Impedances;
procedure Solve_Equations_12_and_13;

implementation

function LOG_km (km: integer; w: extended):extended;
begin
    LOG_km := exp(km*ln(LOG(w)));
end;

function Reciprocal (x, w_1: extended): extended;
begin
   Reciprocal := 1/(x*x - w_1*w_1)
end;

function WhichFunction (Region,Component,SegNum, l, km: integer;
                        X: extended): extended;
var
   F_AB: extended;
begin
   if Region = Experimental then
      case Component of
         RealData: F_AB := Reciprocal(X,w^[l]) *
                           Poly_Y(Zr_Fit^[SegNum].Order,
                                  Zr_Fit^[Segnum].Coef,
                                  LOG(X*Zfit_Hz/Hz_Factor));
         ImagData:  F_AB := X * Reciprocal(X,w^[l]) *
                            Poly_Y(Zi_Fit^[SegNum].Order,
                                   Zi_Fit^[Segnum].Coef,
                                   LOG(X*Zfit_Hz/Hz_Factor))
      end
   else      { Calculated Region }
      case Component of
         RealData:  case km of
                       1: F_AB := (LOG(X)-LOG(w^[l]))*
                                  Reciprocal(X,w^[l]);
                    else  F_AB := (LOG_km(km,X)-LOG_km(km,w^[l]))*
```

```
                                    Reciprocal(X,w^[1])
                    end;
        ImagData:   case km of
                        0: F_AB := (X-w^[1])*Reciprocal (X, w^[1]);
                        1: F_AB := (X*LOG(X)-w^[1]*LOG(w^[1]))*
                                    Reciprocal(X,w^[1]);
                     else  F_AB := (X*LOG_km(km,X)-w^[1]*
                                    LOG_km(km,w^[1]))*
                                    Reciprocal(X,w^[1])
                    end
     end;
   WhichFunction := F_AB
end;

procedure CheckAbort (var Abort: boolean);
begin
   if (ReadKey = #27) then Abort := TRUE
end;

function AdaptiveQuad (Region,Component,SegmentNum,l,km: integer;
                       var w: Frequency;
                       var Abort: Boolean): extended;
{
   Adaptive Quadrature Algorithm based on Algorithm #4.2:
   R.L. Burden and J.D. Faires, ''Numerical Analysis,''
   (4nd ed.), Prindle, Weber \& Schmidt, Boston, MA, p.187, (1989).
}

var
   Old_APP, New_APP, Tolerance, min, max, h: extended;
   NumDivs: integer;
   Converged: Boolean;

   procedure InitializeQuad;
   begin
      NumDivs := 20;         { Initial value for segment division }
      OLD_APP := 0.0;        { Integral Approximations }
      NEW_APP := 0.0;
      Converged := FALSE;
      if Region = Experimental  { Convergence criteria }
         then Tolerance := 1.0E-8
         else Tolerance := 1.0E-10
   end;
```

```
   function GaussianQuad: extended;
   var
      StepSize, i, j, Index1, Index2: integer;
      Sum, Halved, Meaned, Arg, Int: extended;
   begin
     Sum := 0.0;
     StepSize := MaxSegmentDivs DIV NumDivs;
     for i := 1 to NumDivs do begin
       Index1 := (i-1)*StepSize;
       Index2 := i*StepSize;
       Halved := (w[Index2] - w[Index1])/2.0;
       Meaned := (w[Index2] + w[Index1])/2.0;
       Int := 0.0;
       for j := 1 to GAUSS_ORDER do begin
          ARG := Halved * Tg[j] + Meaned;
          Int := Int + Ag[j] * WhichFunction(Region,Component,
                SegmentNum,1,km,ARG)
          end;
       Sum := Sum + Halved * Int
       end;
     GaussianQuad := Sum
   end;

begin   { Main Program for Adaptive Quadrature }
   InitializeQuad;
   Abort := FALSE;
   Old_APP := GaussianQuad;
   repeat
      NumDivs := NumDivs * 2;
      New_APP := GaussianQuad;
      if ABS(New_APP - Old_APP) < Tolerance then
         Converged := TRUE
      else
         if (NumDivs <> MaxSegmentDivs)
            then Old_APP := New_APP
            else Converged := TRUE;
      if KeyPressed then CheckAbort (Abort);
   until Converged or Abort;
   AdaptiveQuad := NEW_APP;
end;

function ExptLimits (w_l:extended): extended;
begin
   ExptLimits := ln((w^[NumPts]-w_l)*(w^[1]+w_l)/
                    ((w^[NumPts]+w_l)*(w^[1]-w_l)))
end;
```

```
procedure WriteDot;
begin
   write ('.')
end;

procedure Equation_13 (l: integer; var Abort: boolean);
const
   DummyINT = 0;
var
   SegNum, k, m: integer;
   Sum: extended;
begin
   Sum := 0.0;
   SegNum := 0;
   while (NOT Abort) AND (SegNum < Ne) do begin
      INC (SegNum,1);
      Sum := Sum + AdaptiveQuad(Experimental,ImagData,
                  SegNum,l,DummyINT,Expt_w^[SegNum].w, Abort);
      WriteDot
      end;
   if (NOT Abort) then begin     { Continue Calculations }
      B^[l+NewPts] := Zr_oo + (2/PI)*Sum;
      for k := 0 to PolyK do
      A^[l+NewPts,k] := LOG_km (k, w^[l]);
      WriteDot;
      m := 0;
      while (m <= PolyM) AND (NOT Abort) do begin
         Sum := 0.0;  SegNum := 0;
      while (SegNum < Nl) AND (NOT Abort) do begin
         INC (SegNum,1);
         Sum := Sum + AdaptiveQuad(Calculated,ImagData,
                     SegNum,l,m, Low_w^[Segnum].w, Abort);
         end;
         A^[l+NewPts,PolyK+1+m] := -(2/PI)* (Sum - LOG_km(m, w^[l])/
                                    2*ExptLimits (w^[l]));
         WriteDot;
         INC (m,1);
         if KeyPressed then CheckAbort (Abort)
         end;
      end
end;

procedure Equation_12 (l: integer; var Abort: boolean);
const
   DummyINT = 0;
```

```
var
   SegNum, k, m: integer;
   Sum: extended;
begin
   Sum := 0.0;
   SegNum := 0;
   while (NOT Abort) and (SegNum < Ne) do begin
      INC (SegNum);
      Sum := Sum + AdaptiveQuad(Experimental,RealData,SegNum,1,
                                DummyINT,Expt_w^[Segnum].w, Abort);
      WriteDot
      end;
   if Not Abort then  begin   { continue calculations }
      B^[1+NewPts] := -(2*w^[1]/PI)*Sum;        { Get B(1) }
      for m := 0 to PolyM do
          A^[1+NewPts,PolyK+1+m] := LOG_km (m,w^[1]);
      WriteDot;
      k := 0;
      while (k <= PolyK) and (NOT Abort) do begin
         Sum := 0.0; SegNum := 0;
         if k <> 0 then
            while (NOT Abort) and (SegNum < N1) do begin
               INC (SegNum);
               Sum := Sum + AdaptiveQuad(Calculated,RealData,SegNum,
                                         1,k,Low_w^[SegNum].w,Abort);
               end;
         A^[1+NewPts,k] := (2*w^[1]/PI)*(Sum - LOG_km(k,w^[1]) *
                           ExptLimits(w^[1])/(2*w^[1]));
         WriteDot;
         INC(k);
         if KeyPressed then CheckAbort(Abort)
         end;
      end
end;

procedure Create_Frequency_Arrays;
var
   i, j: integer;
   w0, w1, Delta: extended;
begin
   for i := 1 to Ne do begin
      w0 := w^[Index[i-1]];
      w1 := w^[Index[i]];
      Delta := (w1-w0)/MaxSegmentDivs;
      with Expt_w^[i] do begin
         for j := 0 to (MaxSegmentDivs-1) do
```

```
            w[j] := w0 + j*Delta;
          w[MaxSegmentDivs] := w1
          end
       end;
   for i := 1 to N1 do begin
       w0 := Low_AB[i-1];
       w1 := Low_AB[i];
       Delta := (w1 - w0)/MaxSegmentDivs;
       with Low_w^[i] do begin
          for j := 0 to (MaxSegmentDivs-1) do
              w[j] := w0 + j*Delta;
          w[MaxSegmentDivs] := w1
          end
       end
end;

procedure Get_A_Matrix_and_B_vector(var Abort: boolean);
var
   l, i, j: integer;
begin
   new (Expt_w);   new (Low_w);
   Create_Frequency_Arrays;
   WriteXY (10,16, 'Number of points to calculate:  ');
   Write (NewPts);
   WriteXY (10,18, 'Calculating point #          :  ');
   l := -NewPts;
   while (NOT Abort) AND (l<0) do begin
      WriteBlank (42,18);  write ((l+NewPts+1):2);
      if (l mod 2 = 0)
         then Equation_13 (l, Abort)
         else Equation_12 (l, Abort);
      if Keypressed then CheckAbort (Abort);
      INC (l)
      end;
   dispose (Expt_w); dispose (Low_w)
end;

procedure Display_Results;
var
   i, l: integer;

   procedure Header;
   begin
      ClrScr;
      Center (2, 'CALCULATED LOW FREQUENCY IMPEDANCE');
      Center (3, '----------------------------------------');
```

```
      WriteXY (1,4, '   Frequency, Hz            Real Z, Ohms'+
                    '            Imaginary Z, Ohms');
      GotoXY (1,6)
   end;

begin
   for i := 0 to PolyK do
       Zr_Coef[i] := Poly_Coef[i];
   for i := 0 to PolyM do
       Zi_Coef[i] := Poly_Coef[PolyK+1+i];
   Header;
   for l := -NewPts to 0 do begin
       Real_Z^[l] := Poly_Y(PolyK, Zr_Coef, LOG(w^[l]));
       Imag_Z^[l] := Poly_Y(PolyM, Zi_Coef, LOG(w^[l]));
       writeln (w_Hz^[l]:16:5, Real_Z^[l]:25:5, Imag_Z^[l]:27:5);
       if ((l+NewPts+1) mod 15 = 0) then begin
          AnyKey ('More data ... press any key to continue ... ');
          Header
          end
       end;
   AnyKey ('Press any key to continue ...')
end;

procedure Plot_Restored_Impedances;
var
   l: integer;
begin
   Display_Results;
   Xlabel := 'Frequency, Hz';
   Title  := '';
   Ylabel := 'Real, Ohms';
   GraphResults (w_Hz, Real_Z, -NewPts, NumPts, NotLined, Rect);
   SmallHorizText;
   GraphPause (Xp(0.05), Yp(0.95), 1);
   Ylabel := 'Imaginary, Ohms';
   GraphResults (w_Hz, Imag_Z, -NewPts, NumPts, NotLined, Rect);
   SmallHorizText;
   GraphPause (Xp(0.05), Yp(0.95), 1);
   RestoreCRTMode
end;

procedure Solve_Equations_12_and_13;
var
   Abort: boolean;
begin
   Abort := FALSE;
```

```
   AllocateGaussArrays;
   Initialize_variables;
   Get_A_matrix_and_B_vector (Abort);
   if NOT Abort then begin
      Gaussian_Elimination (NewPts-1);
      Plot_Restored_Impedances
      end;
   DeallocateGaussArrays
end;

end.
```

B.4 Input Parameters

```
unit InputKK;
{
   Input parameters for the KK-algorithm
}

{$N+,E+}

interface

uses  Crt, KKDecl;

procedure InputParameters (var Abort: Boolean);

implementation

procedure ReturnString (Prompt: String; var InputString: Str20;
                        var Abort: Boolean);
begin
   gotoxy(10,24); ClrEOL;
   if BeepON then Beeper;
   write (Prompt);
   GetString (InputString, Abort);
end;

procedure Put_Arrow (Y: integer);
begin
   gotoXY (58,Y);
   write (Arrow)
end;
```

```
procedure EXTInput (Column, Row: Integer; Prompt: String;
                    var Changed_Var, Abort: Boolean;
                    var  ExtVar: extended);
var
   ExtString: Str20;
   Error    : integer;
begin
   Put_Arrow (Row);
   repeat
      ReturnString (Prompt, ExtString, Abort);
      if ExtString = '' then
         Error := 0
      else  begin
         Changed_Var := TRUE;
         VAL (ExtString, ExtVar, Error)
         end
   until Error = 0;
   WriteBlank (Column, Row); write (ExtVar:10:7)
end;

procedure IntInput (Column, Row: Integer; Prompt: String;
                    var Changed_Var, Abort: Boolean;
                    var  INTVar: integer);
var
   INTString: Str20;
   Error    : integer;
begin
   Put_Arrow (Row);
   repeat
      ReturnString (Prompt, IntString, Abort);
      if INTString = '' then
         Error := 0
      else  begin
         Changed_Var := TRUE;
         VAL (INTString, INTVar, Error)
         end
   until Error = 0;
   WriteBlank (Column, Row); write (INTVar)
end;

procedure InputScreen;
begin
   ClrScr;
   Center (2, 'Parameters for KK-Algorithm');
   Center (3, '-------------------------------');
   WriteXY (15, 6, 'Polynomial Orders');
```

```
   WriteXY (19, 7, 'Real Impedance Component :  ');
   WriteXY (19, 8, 'Imag Impedance Component :  ');
   WriteXY (15,10, 'Lowest Frequency (Hz)       :  ');
   WriteXY (15,12, 'Low Frequency segments      :  ');
   if NewData then begin
      WriteXY (15,15, 'Automatic/Manual Regression  :  ');
      WriteXY (15,17, 'Number experimental segments :  ')
      end
end;

procedure WriteAutoRegress;
begin
   WriteBlank (47,15);
   if AutoRegress then write('Automatic') else write('Manual')
end;

procedure PrintOldValues;
begin
   gotoXY (47, 7);  write (PolyK);
   gotoXY (47, 8);  write (PolyM);
   gotoXY (47,10);  write (w_0_Hz:10:7);
   gotoXY (47,12);  write (N1);
   if NewData then  begin
      WriteAutoRegress;
      gotoXY (47,17);
      if AutoIndex
         then write (Ne)
         else write ('User-Index')
      end
end;

procedure Initialize_Changed_Variables;
begin
   if NewData then  begin
      Changed_PolyOrder := TRUE;
      Changed_w0 := TRUE
      end
   else  begin
      Changed_PolyOrder := FALSE;
      Changed_w0 := FALSE
      end
end;

procedure EnterParameters (var Abort: boolean);
var
   TempStr, Prompt: String;
```

```
   Choice: char;
   Dummy: boolean;
begin
   Initialize_Changed_Variables;
   INTInput (47,7,'Enter Polynomial Order for REAL component:  ',
            Changed_PolyOrder, Abort, PolyK);
   if NOT Abort then
      INTInput(47,8,'Enter Polynomial Order for IMAG component:  ',
               Changed_PolyOrder, Abort, PolyM);
   if NOT Abort then
      EXTInput(47,10,'Enter Lowest frequency for Calculated ' +
                     'region:  ',Changed_w0, Abort, w_0_Hz);
   if NOT Abort then
      INTInput (47,12, 'Number integration segments in ' +
                       'calculated region:  ',Changed_w0, Abort, N1);
   if NewData AND (NOT Abort) then begin
      Put_Arrow (15);
      YNA('Automatic Least squares regression <Y/N>?  ',
          Choice, Abort);
      case Choice of
         'Y':  AutoRegress := TRUE;
         'N':  AutoRegress := FALSE;
         '^':  { Do Nothing }
      end;
      if NOT Abort then begin
         WriteAutoRegress;
         Put_Arrow (17);
         YNA('User-specified regression indexing <Y/N>?  ',
             Choice, Abort);
         case Choice of
            'Y': begin
                    AutoIndex := FALSE;
                    WriteXY (47, 17, 'User-Index');
                    WriteBlank (WhereX, WhereY)
                 end;
            'N': begin
                    AutoIndex := TRUE;
                    WriteBlank (47, WhereY);
                    STR (MaxSegments, TempStr);
                    Prompt := 'Enter number of regression segments ' +
                              '(<= ' + TempStr + '):  ';
                    repeat
                      INTInput (47, 17, Prompt, Dummy, Abort, Ne)
                    until (Ne <= MaxSegments);
                 end;
            '^': WriteBlank (58, 17);   { DO NOTHING }
```

```
        end  { case }
        end
      end
end;

procedure Calculate_New_Algorithm_Parameters;
begin
   NewPts := PolyK + PolyM + 2;
   Hz_Factor := 2*PI*(AntiLog(ROUND(ABS(Log(w_0_Hz)))))
end;

procedure DefaultParameters;
begin
   Initialize_Changed_Variables;
   PolyK := 9;  PolyM := 9;
   w_0_Hz := AntiLOG((LOG(w_Hz^[1]) - 3));
   AutoRegress := TRUE;
   FastAuto    := TRUE;
   AutoIndex   := FALSE;
   Ne := 8;
   Nl := 7;
   HaveParameters := TRUE
end;

procedure InputParameters (var Abort: boolean);
var
   Choice: char;
begin
   InputScreen;
   if NOT HaveParameters then DefaultParameters;
   PrintOldValues;
   repeat
      YNA ('Any corrections (Y/N)? ', Choice, Abort);
      if Choice = 'Y' then EnterParameters (Abort)
   until ((Choice = 'N') or (Choice = 'A')) or Abort;
   Calculate_New_Algorithm_Parameters;   { NewPts & Hz_Factor }
end;

end.
```

B.5 Graphics Subroutine

```
unit GrafKK;
{  This subprogram contains plotting instructions.  }
```

```
{$N+,E+}

interface

uses    Graph, Crt, KKDecl, Drivers, Fonts;

const   Xaxis = 0;
        Yaxis = 1;
        NotLined = 0;
        Lined    = 1;

type    Figures = (NoSymbol, Rect, Circ, Triangle, Diamond, Point);

var     Xmin, Xmax, X_tic_value,
        Ymin, Ymax, Y_tic_value      : extended;
        X_tic_space, Y_tic_space,
        GraphDriver, GraphMode,
        CRTMaxX, CRTMaxY,
        CRTXmin, CRTXmax,
        CRTYmin, CRTYmax,
        VMajTicLen, VMinTicLen,
        Xpt, Ypt, No_XLabels,
        HMajTicLen, HMinTicLen       : integer;
        LogXaxis, LogYaxis           : boolean;
        AutoPlot                     : boolean;
        first, last                  : integer;
        MaxColor                     : word;
        Title, Xlabel, Ylabel        : str40;


function  Xp (f:real): integer;
function  Yp (f:real): integer;
procedure Auto_plot_parameters (X, Y: DataPtr; first, last: integer);
procedure DrawFigure (FigType: Figures);
procedure DrawTicMarks;
procedure Get_Xpt_Ypt (X, Y: real);
procedure GraphPause (Xpos, Ypos, Mode: integer);
procedure GraphResults (var X, Y: DataPtr;
                        First, Last, Liner: integer;
                        FigType: Figures);
procedure InitializeGraphics;
procedure LabelGraph;
procedure Min_Max (X: DataPtr; first, last: integer;
                   LogAxis: Boolean;  var Min, Max: real);
procedure Nyquist_Plot (first, last, LineType: integer;
                        FigType: Figures);
```

```
procedure Plot_ImagZ_data (first, last, LineType: integer;
                           FigType: Figures);
procedure Plot_RealZ_Data (first, last, LineType: integer;
                           Figtype: Figures);
procedure PlotXY (var X, Y: DataPtr; First, Last, Liner: integer;
                  FigType: Figures);
procedure ReplyBox (X1, Y1, X2, Y2: integer);
procedure SmallHorizText;
procedure UserToCRTUnits (X, Y: real);
procedure XYFigure (X, Y: real;  FigType: Figures);

implementation

const  CGA_Small     = 4;
       EGA_VGA_Small = 5;
       CGA_Label     = 6;
       EGA_VGA_Label = 8;
       CGA_Log       = 4;
       EGA_Log       = 6;

var    SmallFontSize, LabelSize, LogSize, LogPow: integer;

function  Xp (f: real): integer;
begin
  Xp := ROUND (CRTMaxX*f)
end;

function  Yp (f: real): integer;
begin
  Yp := ROUND (CRTMaxY*f)
end;

procedure Abort(Msg : string);
begin
  Writeln (Msg, ': ', GraphErrorMsg(GraphResult)); Halt(1);
end;

procedure Load_Drivers;
begin   { Register all the drivers }
  if RegisterBGIdriver(@CGADriverProc) < 0 then Abort('CGA');
  if RegisterBGIdriver(@EGAVGADriverProc) < 0 then Abort('EGA/VGA');
  if RegisterBGIdriver(@HercDriverProc) < 0 then Abort('Herc');
  if RegisterBGIdriver(@ATTDriverProc) < 0 then Abort('AT&T');
  if RegisterBGIdriver(@PC3270DriverProc) < 0 then Abort('PC 3270');
  if RegisterBGIdriver(@IBM8514DriverProc) < 0 then Abort('PC 3270');
end;
```

```
procedure Load_Fonts;
begin   { Register all the fonts }
  if RegisterBGIfont(@GothicFontProc) < 0 then Abort('Gothic');
  if RegisterBGIfont(@SansSerifFontProc) < 0 then Abort('SansSerif');
  if RegisterBGIfont(@SmallFontProc) < 0 then Abort('Small');
  if RegisterBGIfont(@TriplexFontProc) < 0 then Abort('Triplex');
end;

procedure SpecifyGraphicsDriver;
var
   Hardware: integer;
begin
   TextAttr := Yellow + Blue*16;   ClrScr;
   Center (3, '=========================');
   Center (4, '|    GRAPHICS OPTIONS    |');
   Center (5, '=========================');
   Center (8, 'Specify the graphics hardware of your computer');
   WriteXY (20,10, '1) AT&T 640x400 line card');
   WriteXY (20,11, '2) IBM 8514 graphics adapter');
   WriteXY (20,12, '3) Automatic Detection of graphics driver');
   WriteXY (20,13, '   (will detect CGA, VGA, EGA adapters ...');
   WriteXY (20,15, 'Enter choice:  ');
   repeat read(Hardware) until Hardware in [1, 2, 3];
   Load_Drivers;
   Load_Fonts;
   case Hardware of
        1:  begin
               GraphDriver := ATT400;
               GraphMode := ATT400Hi
            end;
        2:  begin
               GraphDriver := IBM8514;
               GraphMode := IBM8514Hi
            end;
        3:  GraphDriver := Detect
   end
end;

procedure InitializeGraphics;
var
   ErrorCode: integer;
begin
   SpecifyGraphicsDriver;
   case GraphDriver of
      CGA: begin
```

```
            SmallFontSize := CGA_Small;
            LabelSize     := CGA_Label;
            LogSize       := CGA_Log;
            LogPow        := LogSize-1
         end;
    else  begin
            SmallFontSize := EGA_VGA_Small;
            LabelSize     := EGA_VGA_Label;
            LogSize       := EGA_Log;
            LogPow        := LogSize-2;
         end
  end;
  InitGraph (GraphDriver, GraphMode, '');
  ErrorCode := GraphResult;
  CRTMaxX := GetMaxX;    CRTMaxY := GetMaxY;
  CRTXmin := Xp(0.20);   CRTXmax := Xp(0.95);
  CRTYmin := Yp(0.85);   CRTYmax := Yp(0.10);
  VMajTicLen := Yp(0.04);   VMinTicLen := VMajTicLen DIV 2;
  HMajTicLen := Xp(0.025);  HMinTicLen := HMajTicLen DIV 2;
  MaxColor  := GetMaxColor;
  RestoreCRTMode;
end;

procedure SmallHorizText;
begin
  SetTextJustify(LeftText, CenterText);
  SetTextStyle(SmallFont, HorizDir, SmallFontSize);
end;

procedure GraphPause (Xpos, Ypos, Mode: integer);
begin
  if BeepON then Beeper;
  SmallHorizText;
  if Mode = 1 then begin
    OutTextXY (Xpos, Ypos, 'Press <CR> to continue ...');
    CRKey ('')
    end
  else begin
    OutTextXY (Xpos, Ypos, 'Please wait ...');
    delay (1500)
    end
end;

procedure ReplyBox (X1, Y1, X2, Y2: integer);
begin
  SetFillStyle (SolidFill, GetMaxColor);
```

```
   Rectangle (X1, Y1, X2, Y2);
   FloodFill (X1+1, Y1-1, GetMaxColor)
end;

procedure Min_Max ( X: DataPtr; first, last: integer;
                    LogAxis: Boolean;
                    var   Min, Max: real);
var
   J: integer;
begin
   if LogAxis then
      begin Min := ABS(X^[first]); Max := ABS(X^[first]); end
   else
      begin Min := X^[first]; Max := X^[first]; end;
   for J := (first + 1) to last do begin
       if LogAxis then begin
          if ABS(X^[J]) < Min
             then Min := ABS(X^[J])
             else if ABS(X^[J]) > Max then Max := ABS(X^[J])
          end
       else begin
          if X^[J] < Min
             then Min := X^[J]
             else if X^[J] > Max then Max := X^[J]
          end
       end  { FOR }
end;   { Procedure Min_Max }

procedure Linear_Min_Max (Min, Max: extended;  Tic_space: integer;
                          var  Axis_min, Axis_max,
                               Axis_tic_value: extended);
var
   adder, Min_integer, Max_integer: integer;
   Factor: extended;
begin
   if Min <> 0.0 then  Min_integer := Round (LOG(ABS(Min)) + 0.51);
   if Max <> 0.0 then  Max_integer := Round (LOG(ABS(Max)) + 0.51);
   if Min =  0.0 then  Min_integer := Max_integer;
   if Max =  0.0 then  Max_integer := Min_Integer;
   if Min_integer >= Max_integer
      then Factor := AntiLOG(Min_integer - 1)
      else Factor := AntiLOG(Max_integer - 1);
   Axis_tic_value := Factor/10.0;
   Axis_min := ROUND(Min/Factor)*Factor;
   while Axis_min > Min do
      Axis_min := Axis_min - Tic_Space * Axis_tic_value;
```

```
   Axis_max := ROUND(Max/Factor)*Factor;
   while Axis_max < Max do
      Axis_max := Axis_max + Tic_Space * Axis_tic_value;

   Axis_min := INT(Axis_min*10000)/10000;
   Axis_max := INT(Axis_max*10000)/10000;
end;  { Procedure Linear_min_max }

procedure Log_Min_Max (Min, Max: extended;
                       var Axis_min, Axis_max: extended);
begin
   Axis_Min := INT(LOG(Min));
   if LOG(Min) < 0.0 then Axis_Min := Axis_Min - 1.0;
   Axis_Max := INT(LOG(Max));
   if LOG(Max) > 0.0 then Axis_Max := Axis_Max + 1.0
end;

procedure Auto_plot_parameters (X, Y: DataPtr; first,last: integer);
var
   Min_x, Max_x, Min_y, Max_y: real;
   NumXlabels, NumYlabels: integer;
begin
   X_tic_space := 5;
   Y_tic_space := 5;
   Min_Max (X, first, last, LogXaxis, Min_X, Max_X);
   Min_Max (Y, first, last, LogYaxis, Min_Y, Max_Y);
   if LogXaxis then
      Log_Min_Max (Min_X, Max_X, Xmin, Xmax)
   else begin
      Linear_Min_max (Min_X, Max_X, X_tic_space,
                      Xmin, Xmax, X_tic_value);
      repeat
         NumXlabels := ROUND((Xmax - Xmin)/
                             (X_tic_value*X_tic_space)) + 1;
         if NumXlabels > 11 then begin
             X_tic_value := X_tic_value*2;
             Xmax := Xmin;
             repeat
                Xmax := Xmax +  X_tic_Space * X_tic_value
             until Xmax > Max_X
             end;
      until NumXlabels <= 11
      end;
   if LogYaxis then
      Log_Min_Max ( Min_Y, Max_Y, Ymin, Ymax)
```

```
   else begin
      Linear_Min_max (Min_y, Max_y, Y_tic_space,
                      Ymin, Ymax, Y_tic_value);
      NumYlabels := ROUND((Ymax - Ymin)/
                          (Y_tic_value*Y_tic_space)) + 1;
      if NumYlabels > 11 then begin
         Y_tic_value := Y_tic_value*2;
         Ymax := Ymin;
         repeat
            Ymax := Ymax + Y_tic_Space * Y_tic_value
         until Ymax > Max_Y
         end;
      if ABS((Min_y-Ymin)/Y_tic_value) < 0.50 then
         Ymin := Ymin -  Y_tic_space * Y_tic_value;
      if ABS((Max_y-Ymax)/Y_tic_value) < 0.50 then
         Ymax := Ymax +  Y_tic_space * Y_tic_value
      end
end;   { Procedure Auto_plot_parameters }

procedure UserToCRTUnits (X, Y: real);
begin
   Xpt := round(CRTXmin + (X - Xmin)/
               (Xmax - Xmin)*(CRTXmax - CRTXmin));
   Ypt := round(CRTYmin + (Y - Ymin)/
               (Ymax - Ymin)*(CRTYmax - CRTYmin));
end;

procedure XMajorTics (X: real; Decimal: boolean);
var
   XticLabel  : LongInt;
   MajTicLabel: string[10];
begin
   UserToCRTUnits (X, Ymin);
   Line (Xpt, Ypt, Xpt, Ypt - VMajTicLen);
   UserToCRTUnits (X, Ymax);
   Line (Xpt, Ypt, Xpt, Ypt + VMajTicLen);
   if Decimal then
      STR (X:7:2, MajTicLabel)
   else begin
      XticLabel := ROUND(X);
      STR (XticLabel, MajTicLabel)
      end;
   if LogXaxis then begin
      SetTextJustify(CenterText, CenterText);
      SetTextStyle (SmallFont, HorizDir, LogSize);
      OutTextXY(Xpt, CRTYmin + 15, '10');
```

```
      SetTextJustify (LeftText, CenterText);
      SetTextStyle (SmallFont, HorizDir, LogPow);
      OutTextXY(Xpt + 11, CRTYmin + 10, MajTicLabel)
      end
   else
      OutTextXY(Xpt, CRTYmin + 10, MajTicLabel)
end;

procedure XMinorTics (X: real);
begin
   UserToCRTUnits (X, Ymin);
   Line (Xpt, Ypt, Xpt, Ypt - VMinTicLen);
   UserToCRTUnits (X, Ymax);
   Line (Xpt, Ypt, Xpt, Ypt + VMinTicLen);
end;

procedure YMajorTics (Y: real; Decimal: Boolean);
var
   YticLabel  : LongInt;
   MajTicLabel: string[10];
begin
   UserToCRTUnits (Xmin, Y);
   Line (Xpt, Ypt, Xpt + HMajTicLen, Ypt);
   UserToCRTUnits (Xmax, Y);
   Line (Xpt, Ypt, Xpt - HMajTicLen, Ypt);
   if Decimal then
      STR (Y:10:2, MajTicLabel)
   else  begin
      YticLabel := ROUND(Y);
      STR (YticLabel, MajTicLabel)
      end;
   if LogYaxis then begin
      SetTextJustify(RightText, CenterText);
      SetTextStyle (SmallFont, HorizDir, LogSize);
      OutTextXY(CRTXMin - 20, Ypt, '10');
      SetTextJustify (LeftText, CenterText);
      SetTextStyle (SmallFont, HorizDir, LogPow);
      OutTextXY(CRTXMin - 18, Ypt - 9, MajTicLabel)
      end
   else
      OutTextXY(CRTXMin - 10, Ypt, MajTicLabel)
end;

procedure YMinorTics (Y: real);
begin
   UserToCRTUnits (Xmin, Y);
```

```
   Line (Xpt, Ypt, Xpt + HMinTicLen, Ypt);
   UserToCRTUnits (Xmax, Y);
   Line (Xpt, Ypt, Xpt - HMinTicLen, Ypt);
end;

procedure LinearTics (Min, Max, Tic_value: extended;
                      Tic_space, Axis:integer);
var
   I: extended;
   J, MajorIntervals, NumLoops: integer;
   MajorTic, Decimal: Boolean;
begin
   I := Min;
   NumLoops := ROUND((Max-Min)/(Tic_value * Tic_space));
   if Tic_Value*Tic_Space < 5.0
      then Decimal := TRUE
      else Decimal := False;
   for MajorIntervals := 1 to NumLoops do begin
       MajorTic := TRUE;
       for J := 1 to Tic_Space do begin
           if MajorTic then  begin
              case Axis of
                Xaxis:  XMajorTics (I, Decimal);
                Yaxis:  YMajorTics (I, Decimal)
              end;
              MajorTic := FALSE
              end
           else
              case Axis of
                 Xaxis:  XMinorTics (I);
                 Yaxis:  YMinorTics (I)
              end;
           I := I + Tic_Value
           end
       end;
   case Axis of    { Maximum Tic mark and Label }
      Xaxis:  XMajorTics (Max, Decimal);
      Yaxis:  YMajorTics (Max, Decimal)
   end;
end;

procedure LogTics (LogMin, LogMax, Axis: integer);
var
   LogI, Loop: integer;
   Low, High, J, LogJ, Delta: extended;
   MajorTic: Boolean;
```

```
begin
   for LogI := LogMin to (LogMax - 1) do begin
       Low  := AntiLog (LogI);    J := Low;
       High := AntiLog (LogI+1); Delta := High/10.;
       MajorTic := TRUE;  Loop := 0;
       repeat
          Loop := Loop + 1;
          LogJ := LOG(J);
          if MajorTic then begin
             case Axis of
                Xaxis:  XMajorTics (LogJ, False);
                Yaxis:  YMajorTics (LogJ, False)
             end;
             MajorTic := FALSE
             end
          else
             case Axis of
                Xaxis:  XMinorTics (LogJ);
                Yaxis:  YMinorTics (LogJ)
             end;
          J := J + Delta;
       until Loop = 9
       end;
   case Axis of
      Xaxis:  XMajorTics (LogMax, False);
      Yaxis:  YMajorTics (LogMax, False)
   end
end;

procedure DrawTicMarks;
begin
   if LogXaxis then       (*  Draw X-Axis tic marks  *)
      LogTics (TRUNC(Xmin), TRUNC(Xmax), Xaxis)
   else begin
      SetTextJustify(CenterText, TopText);
      SetTextStyle (SmallFont, Horizdir, SmallFontSize);
      LinearTics (Xmin, Xmax, X_tic_value, X_tic_space, Xaxis)
      end;
   if LogYaxis then       (*  Draw Y-axis tic marks  *)
      LogTics (TRUNC(Ymin), TRUNC(Ymax), Yaxis)
   else begin
      SetTextJustify (RightText, CenterText);
      SetTextStyle (SmallFont, HorizDir, SmallFontSize);
      LinearTics (Ymin, Ymax, Y_tic_value, Y_tic_space, Yaxis)
      end
end;
```

```
procedure LabelGraph;
var
   X1, Y1, Y2, X3, Y3: integer;
begin
   SetTextJustify (CenterText, CenterText);
   SetTextStyle (SmallFont, HorizDir, LabelSize);
   X1 := (CRTXmax + CRTXmin) DIV 2;
   Y1 := CRTYmax DIV 2;
   OutTextXY (X1, Y1, Title);

   Y2 := CRTYmin + (2*(CRTMaxY - CRTYmin) DIV 3);
   OutTextXY (X1, Y2, XLabel);

   X3 := CRTXmin DIV 4;
   Y3 := (CRTYMax + CRTYmin) DIV 2;
   SetTextStyle (SmallFont, VertDir, LabelSize);
   OutTextXY (X3, Y3, YLabel)
end;

procedure DrawFigure (FigType: Figures);
var
   X1, X3, Y1, Y2, Y4: integer;
begin
   Case ORD(Figtype) of
      1:  Rectangle (Xpt-2, Ypt-2, Xpt+2, Ypt+2);
      2:  Circle (Xpt, Ypt, 5);
      3:  begin   { Triangle }
            X1 := Xpt - 3;
            Y1 := Ypt + 3;
            Y2 := Ypt - 3;
            X3 := Xpt + 3;
            Line (X1, Y1, Xpt, Y2);
            Line (Xpt, Y2, X3, Y1);
            Line (X3, Y1, X1, Y1)
          end;
      4:  begin          { Diamond }
            X1 := Xpt - 2;
            Y2 := Ypt + 2;
            X3 := Xpt + 2;
            Y4 := Ypt - 2;
            Line (X1, Ypt, Xpt, Y2);
            Line (Xpt, Y2, X3, Ypt);
            Line (X3, Ypt, Xpt, Y4);
            Line (Xpt, Y4, X1, Ypt)
          end;
```

```
      5:  PutPixel (Xpt, Ypt, MaxColor);
   end;
end;

procedure Get_Xpt_Ypt (X, Y: real);
begin
   if LogXaxis then begin
      if LogYaxis
         then UserToCRTUnits (LOG(X), LOG(Y))
         else UserToCRTUnits (LOG(X), Y)
      end
   else begin
      if LogYaxis
         then UserToCRTUnits (X, LOG(Y))
         else UserToCRTUnits (X, (Y))
      end
end;

procedure XYFigure (X, Y: real;  FigType: Figures);
begin
   Get_Xpt_Ypt (X, Y);
   if FigType <> NoSymbol then DrawFigure (FigType)
end;

procedure PlotXY (var X, Y: DataPtr; First, Last, Liner: integer;
                  FigType: Figures);
var
   i: integer;
   X1, Y1: integer;
   TempFig: Figures;
begin
   for i := first to last do begin
       Get_Xpt_Ypt (X^[i], Y^[i]);
       if FigType <> NoSymbol then begin
          if (i<=0)
             then TempFig := Triangle
             else TempFig := FigType;
          DrawFigure (TempFig)
          end;
       if Liner = Lined then begin
          if i > first then Line (X1, Y1, Xpt, Ypt);
          X1 := Xpt;
          Y1 := Ypt
          end
   end
end;
```

```
procedure DrawDemarcationLine (X0: real);
var
   X1, Y1: integer;
begin
   SetLineStyle (3, 0, 1);
   Get_Xpt_Ypt (X0, Ymin);
   X1 := Xpt;  Y1 := Ypt;
   Get_Xpt_Ypt (X0, Ymax);
   Line (X1, Y1, Xpt, Ypt);
   SetLineStyle (0, 0, 1);
   SmallHorizText;
   OutTextXY (Xp(0.25), Yp(0.18), 'Calculated');
   SetTextJustify(RightText, CenterText);
   OutTextXY (Xp(0.90), Yp(0.18), 'Experimental')
end;

procedure GraphResults (var X, Y: DataPtr;
                        First, Last, Liner: integer;
                        FigType: Figures);
begin
   SetGraphMode(GraphMode);
   ClearDevice;
   if MaxColor > 1 then begin
      SetBkColor(Blue);
      SetColor(Yellow)
      end;
   SetViewPort (0, 0, CRTMaxX, CrtMaxY, ClipON);
   Rectangle (0, 0, CRTMaxX, CrtMaxY);
   Rectangle (CRTXmin, CRTYmin, CRTXmax, CRTYmax);
   Auto_Plot_Parameters (X, Y, First, Last);
   DrawTicMarks;
   if (First < 0) then DrawDemarcationLine (X^[0]);
   LabelGraph;
   PlotXY (X, Y, First, Last, Liner, FigType)
end;

procedure Plot_RealZ_Data (first, last, LineType: integer;
                           Figtype: Figures);
begin
   Xlabel := 'Frequency,  Hz';
   Ylabel := 'Real, Z ''';
   LogXaxis := TRUE;
   if first = 0 then
      GraphResults (w_Hz, Real_Z, 1, Last, LineType, FigType)
   else begin
```

```
      GraphResults (w_Hz, Real_Z, first, Last, LineType, FigType);
      GraphPause (Xp(0.25), Yp(0.75), 2)
      end;
   Title := ''
end;

procedure Plot_ImagZ_data (first, last, LineType: integer;
                           FigType: Figures);
begin
   Xlabel := 'Frequency,  Hz';
   Ylabel := 'Imaginary,  -Z"';
   LogXaxis := TRUE;
   if first = 0 then
      GraphResults (w_Hz, Imag_z, 1, last, LineType, FigType)
   else begin
      GraphResults (w_Hz, Imag_z, first, last, LineType, FigType);
      GraphPause (Xp(0.25), Yp(0.75), 2)
      end;
   Title := ''
end;

procedure Nyquist_Plot (first, last, LineType: integer;
                        FigType: Figures);
begin
   Title := '';
   Xlabel := 'Real,  Z''';
   Ylabel := 'Imaginary,  -Z"';
   LogXaxis := FALSE;
   SetGraphMode(GraphMode);
   ClearDevice;
   if MaxColor > 1 then begin
      SetBkColor(Blue);
      SetColor(Yellow)
      end;
   SetViewPort (0, 0, CRTMaxX, CrtMaxY, ClipON);
   Rectangle (0, 0, CRTMaxX, CrtMaxY);
   Rectangle (CRTXmin, CRTYmin, CRTXmax, CRTYmax);
   Auto_Plot_Parameters (Real_Z, Imag_Z, first, last);
   while XMax < ZrMax.Zr do
      Xmax := Xmax + X_tic_value*X_tic_space;
   DrawTicMarks;
   LabelGraph;
   PlotXY (Real_Z, Imag_Z, First, Last, LineType, FigType);
   XYFigure (Zi0.Zr, Zi0.Zi, Rect);
   XYFigure (ZrMax.Zr, ZrMax.Zi, Rect);
   XYFigure (Zi0.Zr, Zi0.Zi, Circ);
```

```
   XYFigure (ZrMax.Zr, ZrMax.Zi, Circ)
end;

end.
```

B.6 File Input/Output

```
unit Files;
{
  Performs file handler routines ... retrieval and storage
  of impedance files using Z-PLOT format
}

{$N+}

interface

uses  DOS, Crt, KKDecl;

procedure Retrieve_Data (var AbortIO: boolean);
procedure RetrieveDataFile (var AbortIO: boolean);
procedure Store_Results;

implementation

procedure IO_Error (DirPath: string; FN: Str20; ErrorCode: integer);
var
   DirFile, Error: string;
begin
   Window (1,7,80,21);  ClrScr;
   Blinking_Label ('******  I/O ERROR  ******');
   DirFile := 'DirPath\FileName: ' + DirPath + FN;
   Center (WhereY + 2, DirFile);
   case ErrorCode of
     2  : Error := 'File NOT found .. retype ' +
                  'filename of check disk';
     3  : Error := 'Directory path name NOT found ' +
                  '... retype directory';
     101: Error := 'Disk is FULL ... replace floppy disk!!!';
     106: Error := 'Invalid numeric format ... ' +
                  'probably reading wrong file!'
   end;
   Center (WhereY+2, Error);
   Window (1,1,80,25)
end;
```

```
procedure  GetDirectory (var AbortIO: boolean);
var
   DirInfo: SearchRec;
   DT: DateTime;
   Temp, DirMask: Str20;
   DoDir:  boolean;
   InChar: char;
begin
   DoDir := TRUE;
   DirMask := '*.*';
   Gotoxy (5,24);
   Write ('Directory mask (*.* is default):  ');
   GetString (Temp, AbortIO);
   if Temp <> '' then DirMask := Temp;
   if NOT AbortIO then begin
      Window (1,7,80,21);  ClrScr;
      FindFirst (DirPath+DirMask, Archive+Directory, DirInfo);
      while (DosError = 0) AND (DoDir) do begin
         UnPackTime(DirInfo.Time, DT);
         write ('   ', DirInfo.Name:12);
         if DirInfo.Size = 0
            then write ( '       < DIR >         ')
            else write (DirInfo.Size:12, ' bytes     ');
         writeln (DT.Month:2,'/',DT.Day:2,'/',DT.Year,'   ',
                  DT.Hour:2,':',DT.Min:2,':',DT.Sec:2);
         if (wherey = 13) then begin
            WriteXY (5,15,'More Files ... Any key to continue ' +
                          '...<ESC> to end');
            InChar := ReadKey;
            if (InChar = #27) then DoDir := FALSE else ClrScr;
            end;
         FindNext(DirInfo)
         end;
      Window (1,1,80,25)
      end;
end;

procedure InputDirPath (var DirPath: String; var AbortIO: boolean);
var
   i, j: integer;
   InChar: char;
   CharArray: array[1..20] of char;
   Done: Boolean;
begin
   repeat
```

```
      Done := FALSE;
      i := 0;
      while (NOT Done) AND (NOT AbortIO) do begin
         InChar := ReadKey;
         case InChar of
            #27: AbortIO := TRUE;
            #8 : begin
                    WriteXY (WhereX-1, WhereY, ' ');
                    GotoXY (WhereX-1, WhereY);
                    if i >= 1 then dec(i)
                 end;
            #13: Done := TRUE;
          else   begin
                    inc(i);
                    CharArray[i] := InChar;
                    WriteXY (WhereX, WhereY, InChar);
                 end
         end
         end;
      if AbortIO
         then GetDir (0, DirPath)      { Get current Directory }
         else for j := 1 to i do
      DirPath := DirPath + CharArray[j];
   until Done OR AbortIO;
end;

procedure GetDirPath (var DirPath: string;
                      var AbortIO: boolean);
var
   i, ErrorCode: integer;
begin
   WriteXY (5, 5, 'Current path name (drive and/or directory):  ');
   write (DirPath);
   YNA ('Change current path name <Y/N>?  ', Choice, AbortIO);
   if Choice = 'Y' then
      repeat
         DirPath := '';
         if BeepON then Beeper;
         WriteXY (5,24, 'Enter new path name:  ');
         InputDirPath (DirPath, AbortIO);
         if NOT AbortIO then begin
            {$I-} ChDir(DirPath);  {$I+}
            ErrorCode := IOResult;
            if ErrorCode <> 0 then
               IO_Error (DirPath, '', ErrorCode)
            else  begin
```

```
            Window (1,7,80,21);  ClrScr; Window (1,1,80,25) end
         end
      until ((DirPath <> '') and (ErrorCode = 0)) or AbortIO;
   if NOT AbortIO then begin
      if DirPath[Length(DirPath)] <> '\' then begin
         DirPath := DirPath + '\';
         for i := 1 to Length(DirPath) do
             DirPath[i] := UpCase (DirPath[i])
         end;
      WriteBlank (50, 5);
      WriteXY (50,5, DirPath)
      end
end;

procedure InputFileName (var FileName: Str20; var AbortIO: boolean);
var
   i, j: integer;
   InChar: char;
   CharArray: array[1..20] of char;
   Done: Boolean;
begin
   repeat
      WriteBlank (5,24);
      WriteXY (5,24,'Enter Filename:  ');
      Done := FALSE;
      i := 0;
      FileName := '';
      while (NOT Done) AND (NOT AbortIO) do begin
         InChar := ReadKey;
         case InChar of
            #27: AbortIO := TRUE;
            #8 : begin
                    WriteXY (WhereX-1, WhereY, ' ');
                    GotoXY (WhereX-1, WhereY);
                    if i >= 1 then dec(i)
                 end;
            #13: Done := TRUE;
           else  begin
                    inc(i);
                    CharArray[i] := InChar;
                    WriteXY (WhereX, WhereY, InChar);
                    if i = 20 then Done := TRUE
                 end
         end
         end;
      if AbortIO
```

```
        then FileName := ''
        else for j := 1 to i do
                FileName := FileName + CharArray[j];
   until (FileName <> '') OR AbortIO;
end;

procedure AssignFileNumber (var DirPath: string; var FileName: Str20;
                            var file1: text; var AbortIO: boolean);
begin
   GetDirPath (DirPath, AbortIO);
   if NOT AbortIO then begin
      YNA ('Request directory (Y/N):  ', Choice, AbortIO);
      if Choice = 'Y' then GetDirectory (AbortIO);
      if NOT AbortIO then begin
         InputFileName (FileName, AbortIO);
         Assign (file1, DirPath+FileName)
         end
      end
end;

procedure GetFileName (Mode: integer; var DirPath: string;
                       var FileName: Str20; var file1: text;
                       var AbortIO: boolean);
var
   ErrorCode: integer;
begin
   repeat
      AssignFileNumber (DirPath, FileName, file1, AbortIO);
      if NOT AbortIO then begin
         {$I-}     (* Suspend error traps *)
         case Mode of
            Retrieve: reset (file1);
            Store   : rewrite (file1)
         end;
         {$I+}
         ErrorCode := IOResult;
         if ErrorCode <> 0 then begin
            IO_Error(DirPath, Filename, ErrorCode);
            GetDir (0, DirPath)      { Get current Directory }
            end
         end
   until (ErrorCode = 0) OR AbortIO;
end;

procedure Invert_Array_Indices;
var
```

```
   i, move: integer;
begin
   for i := 1 to (NumPts div 2) do begin
      move := NumPts - i + 1;
      Swap (w_Hz^[i], w_Hz^[move]);
      Swap (Real_Z^[i], Real_Z^[move]);
      Swap (Imag_Z^[i], Imag_Z^[move]);
      end
end;

procedure Display_Frequency_Limits;
begin
   WriteXY (10, 4, 'Pathname/Filename              :  ');
   write (DirPath+FileName);
   WriteXY (10, 6, 'Number of data points          :  ');
   write (NumPts);
   WriteXY (10, 8, 'Minimum measured Frequency (Hz): ');
   write (w_Hz^[1]:12);
   WriteXY (10,10, 'Maximum measured Frequency (Hz): ');
   write (w_Hz^[NumPts]:12);
   if Inductive then begin
      if BeepON then Beeper;
      WriteXY (10,13,'NOTE:  Data Set possibly contains ' +
                     'an inductive loop!!!');
      WriteXY (17,15,'User will determine the following:');
      WriteXY (22,16,'a) Maximum Real Impedance , and,');
      WriteXY (22,17,'b) Real Impedance intercept ...');
      YesNo ('Perform Capacitive Calculations? ', Choice);
      if Choice = 'Y'
         then InductiveDone := FALSE
         else InductiveDone := TRUE
      end
   else
      CRKey ('Press <Enter> to continue ...')
end;

procedure Perform_Retrieval (var file1: text);
var
   Z, Lag, dummy: real;
   ErrorCode: integer;
begin
   ClrScr;
   Center (2,'******** RETRIEVING DATA ********');
   NumPts := 0;   Inductive := FALSE;
   {$I-}
   while not eof (File1) do
```

```
      begin
      Inc(NumPts);
      readln (file1, w_Hz^[NumPts], Real_Z^[NumPts], Imag_Z^[NumPts],
                     Lag, Z, dummy);
      Imag_Z^[NumPts] := -Imag_Z^[NumPts];
      if Imag_Z^[NumPts] < 0.0 then Inductive := TRUE;
      end;
   {$I+}
   ErrorCode := IOResult;
   if ErrorCode <> 0 then IO_Error(DirPath, FileName, ErrorCode);
   close (file1);
   if w_Hz^[1] > w_Hz^[NumPts] then Invert_Array_Indices;
   Display_Frequency_Limits;
   HaveData := TRUE;  NewData := TRUE;  HaveParameters := FALSE;
end;

procedure Retrieve_data (var AbortIO: boolean);
var
   file1: text;
begin
   ClrScr;
   HaveData := FALSE;
   Center (2, 'RETRIEVE IMPEDANCE DATA FILE');
   Center (3, '-----------------------------------------');
   GetFileName (Retrieve, DirPath, FileName, file1, AbortIO);
   if NOT AbortIO then Perform_Retrieval (file1)
end;

procedure RetrieveDataFile (var AbortIO: boolean);
var
   Choice: char;
begin
   AbortIO := FALSE;
   if HaveData then begin
      ClrScr;
      Center (2, 'DATA FILE TO BE PROCESSED');
      Center (3, '-------------------------------'); writeln;
      WriteXY (15,5, 'Current PathName\DataFile:  ');
      writeln (DirPath+FileName);
      YNA ('Use Current Data File <Y/N>?  ', Choice, AbortIO);
      if Choice = 'N' then Retrieve_Data (AbortIO)
      end
   else
      Retrieve_Data (AbortIO);
end;
```

```
procedure Store_Results;
const
   dummy = 0;
var
   J, ErrorCode, NumColumns, NumLabels: integer;
   file1: text;
   StorePath: string;
   StoreFile: Str20;
   AbortIO: Boolean;
begin
   ClrScr;
   Center (2, 'STORE KK-TRANSFORM RESULTS');
   Center (3, '--------------------------------------');
   writeln;
   GetDir (0, StorePath);
   GetFileName (Store, StorePath, StoreFile, file1, AbortIO);
   if NOT AbortIO then begin
      {$I-}          (* Suspend error traps for writing *)
      for j := NumPts downto -NewPts do
          writeln (file1, w_Hz^[J]:13,' ',
                   Real_Z^[J]:15,' ',-Imag_Z^[J]:15,' ',
                   Arctan(-Imag_Z^[j]/Real_Z^[j])*180/PI:15,' ',
                   Sqrt(SQR(-Imag_Z^[j]) +
                   SQR(Real_Z^[j])):15,' ', dummy);
      {$I+}
      ErrorCode := IOResult;
      if ErrorCode <> 0 then IO_Error (DirPath, FileName, ErrorCode);
      close (file1)
      end
end;

end.
```

B.7 Inductive Data

```
unit Induct;
{
   Subprograms for Inductive Data:
   estimation of Zi_max and Zr_max.
}

{$N+,E+}

interface
```

```
uses  Crt, Graph, KKDecl, GrafKK;

procedure  Inductive_Data;

implementation

function Xo (Yo, X1, Y1, X2, Y2: extended): extended;
var
   m: extended;
begin
   m  := (Y2-Y1)/(X2-X1);
   Xo := (Yo-Y1)/m + X1
end;

function Yo (Xo, X1, Y1, X2, Y2: extended): extended;
var
   m: extended;
begin
   m  := (Y2-Y1)/(X2-X1);
   Yo := m*(Xo-X1) + Y1
end;

function Yval (Ypt: integer): extended;
begin
   Yval := (Ymax - Ymin)*(Ypt - CRTYmin)/
           (CRTYmax - CRTYmin) + Ymin
end;

function Xval (Xpt: integer): extended;
begin
   Xval := (Xmax - Xmin)*(Xpt - CRTXmin)/
           (CRTXmax - CRTXmin) + Xmin
end;

procedure DrawCross (X, Y: extended);
begin
   Get_Xpt_Ypt (X, Y);
   Line (Xpt-4,Ypt,Xpt+4,Ypt);
   Line (Xpt,Ypt-4,Xpt,Ypt+4)
end;

procedure Get_Zi_first_last (var first, last: integer);
var
   i, CR: integer;
   KeyInput: char;
begin
```

```
   Title := 'Z''(w) intercept --- Select Interval';
   Plot_ImagZ_Data (0, NumPts, NotLined, Rect);
   SetWriteMode (XORPut);
   i := 1;  CR := 0;
   while CR <> 2 do begin
      DrawCross (w_Hz^[i], Imag_Z^[i]);
      repeat
         KeyInput := ReadKey
      until KeyInput in [#0, #13];
      if KeyInput = #0 then KeyInput := ReadKey;
      DrawCross (w_Hz^[i], Imag_Z^[i]);   { Erase Cross }
      case KeyInput of
         #13:  begin
                  Circle (Xpt, Ypt, 5);
                  inc(CR);
                  Case CR of
                     1:  First := i;
                     2:  Last  := i
                  end
               end;
         #75:  if i>1 then dec(i);        { Left  arrow }
         #77:  if i<NumPts then INC(i);   { Right arrow }
        else   if BeepON then Beeper
      end  { case }
      end  { while }
end;

procedure Get_Zr_first_last (var first, last: integer);
var
   i, CR: integer;
   KeyInput: char;
begin
   Title := 'Z''(w) maximum --- Select Interval';
   Plot_RealZ_Data (0, NumPts, NotLined, Rect);
   SetWriteMode (XORPut);
   i := 1;  CR := 0;
   while CR <> 2 do begin
      DrawCross (w_Hz^[i], Real_Z^[i]);
      repeat
         KeyInput := ReadKey
      until KeyInput in [#0, #13];
      if KeyInput = #0 then KeyInput := ReadKey;
      DrawCross (w_Hz^[i], Real_Z^[i]);   { Erase Cross }
      case KeyInput of
         #13:  begin
                  Circle (Xpt, Ypt, 5);
```

```
                INC(CR);
                Case CR of
                   1:  First := i;
                   2:  Last  := i
                end
             end;
        #75:  if i>1 then DEC(i);        { Left  arrow }
        #77:  if i<NumPts then INC(i);   { Right arrow }
       else   if BeepON then Beeper
     end  { case }
     end  { while }
end;

procedure IntersectLines (Xpo, Ypo: integer);
begin
   SetLineStyle (DottedLn, 0, NormWidth);
   Line (Xpo, Ypo, Xpo, CRTYmin);
   Line (Xpo, Ypo, Xpo, CRTYmax);
   Line (Xpo, Ypo, CRTXmin, Ypo);
   Line (Xpo, Ypo, CRTXMax, Ypo);
   SetLineStyle (DashedLn, 0, NormWidth);
end;

procedure DrawRect (Xp0, Yp0: integer);
begin
   SetLineStyle (SolidLn, 0, NormWidth);
   Rectangle (Xp0-4, Yp0-4, Xp0+4, Yp0+4)
end;

procedure DrawRectLines (Xp0, Yp0, Xp1, Yp1, Xp2, Yp2: integer);
begin
   DrawRect (Xp0, Yp0);
   SetLineStyle (DashedLn, 0, NormWidth);
   Line (Xp1, Yp1, Xp0, Yp0);
   Line (Xp0, Yp0, Xp2, Yp2)
end;

procedure Get_Zi0;
var
   first, last, i: integer;
   Xp0, Yp0, Xp1, Yp1, Xp2, Yp2: integer;
   Freq, Imag: DataPtr;
   ChInput: char;
begin
   Get_Zi_first_last (first, last);
   LogXaxis := FALSE;
```

```
{  LogXaxis := TRUE; }
   Title := 'Z"(w) = 0 ... select frequency';
   GraphResults (w_Hz, Imag_z, first, last, Lined, Rect);
   SetWriteMode (XORPut);
   Zi0.Zi := 0.0;
   i := last;
   while (Imag_Z^[i] > 0.0) do DEC(i);
   Zi0.w := Xo (Zi0.Zi, w_Hz^[i], Imag_z^[i],
                w_Hz^[i+1], Imag_z^[i+1]);
   Get_Xpt_Ypt (Zi0.w, Zi0.Zi);
   Xp0 := Xpt; Yp0 := Ypt;
   Get_Xpt_Ypt (w_Hz^[i], Imag_Z^[i]);
   Xp1 := Xpt; Yp1 := Ypt;
   Get_Xpt_Ypt (w_Hz^[i+1], Imag_Z^[i+1]);
   Xp2 := Xpt; Yp2 := Ypt;
   repeat
      DrawRectLines (Xp0, Yp0, Xp1, Yp1, Xp2, Yp2);
      IntersectLines (Xp0, Yp0);
      repeat
         ChInput := ReadKey
      until ChInput in [#13, #0];
      if ChInput = #0 then ChInput := ReadKey;
      DrawRectLines (Xp0, Yp0, Xp1, Yp1, Xp2, Yp2);   (* Erase *)
      IntersectLines (Xp0, Yp0);
      case ChInput of
         #75:  DEC(Xp0);   { Left  arrow }
         #77:  INC(Xp0);   { Right arrow }
      end;
   until ChInput = #13;
{  Zi0.w := AntiLog(Xval(Xp0)); }
   Zi0.w := Xval(Xp0);

   LogXaxis := FALSE;
   ClearDevice;
   Title := 'Select Z''(w) at Z"(w) = 0';
   YLabel := 'Real Z, Ohms';
   GraphResults (w_Hz, Real_Z, first, last, Lined, Rect);
   SetWriteMode (XORPut);
   Zi0.Zr := Yo (Zi0.w, w_Hz^[i], Real_Z^[i],
                 w_Hz^[i+1], Real_Z^[i+1]);
   Get_Xpt_Ypt (Zi0.w, Zi0.Zr);
   Xp0 := Xpt; Yp0 := Ypt;
   Get_Xpt_Ypt (w_Hz^[i], Real_Z^[i]);
   Xp1 := Xpt; Yp1 := Ypt;
   Get_Xpt_Ypt (w_Hz^[i+1], Real_Z^[i+1]);
   Xp2 := Xpt; Yp2 := Ypt;
```

```
   SetLineStyle (DashedLn, 0, NormWidth);
   repeat
      DrawRectLines (Xp0, Yp0, Xp1, Yp1, Xp2, Yp2);
      IntersectLines (Xp0, Yp0);
      repeat
         ChInput := ReadKey
      until ChInput in [#13, #0];
      if ChInput = #0 then ChInput := ReadKey;
      DrawRectLines (Xp0, Yp0, Xp1, Yp1, Xp2, Yp2);  (* Erase *)
      IntersectLines (Xp0, Yp0);
      case ChInput of
         #72:  DEC(Yp0);   { Up   arrow }
         #80:  INC(Yp0)    { Down arrow }
      end;
   until ChInput = #13;
   Zi0.Zr := Yval (Yp0);
end;

procedure Get_ZrMax;
var
   first, last, i, MaxI: integer;
   Xp0, Yp0, Xp1, Yp1, Xp2, Yp2: integer;
   Freq, Imag: DataPtr;
   ChInput: char;
begin
   Get_Zr_first_last (first, last);
   LogXaxis := TRUE;
{  LogXaxis := FALSE; }
   SetGraphMode(GraphMode);
   if MaxColor > 1 then begin
      SetBkColor(Blue);
      SetColor(Yellow)
      end;
   SetViewPort (0, 0, CRTMaxX, CrtMaxY, ClipON);
   Rectangle (0, 0, CRTMaxX, CrtMaxY);
   Rectangle (CRTXmin, CRTYmin, CRTXmax, CRTYmax);
   Auto_Plot_Parameters (w_Hz, Real_Z, first, last);
   Ymax := Ymax + 2*(Y_tic_value*Y_tic_space);
   DrawTicMarks;
   Title := 'Select maximum Z''(w)';
   LabelGraph;
   PlotXY (w_Hz, Real_Z, first, last, Lined, Rect);
   SetWriteMode (XORPut);
   MaxI := first;
   for i := (first+1) to last do
       if Real_Z^[i] > Real_Z^[i-1] then MaxI := i;
```

```
   ZrMax.w := (w_Hz^[MaxI] + w_Hz^[MaxI+1])/2.0;
   ZrMax.Zr := Real_Z^[MaxI];
   Get_Xpt_Ypt (ZrMax.w, ZrMax.Zr);
   Xp0 := Xpt;  Yp0 := Ypt;
   repeat
      DrawRect (Xp0, Yp0);
      IntersectLines (Xp0, Yp0);
      repeat
         ChInput := ReadKey
      until ChInput in [#13, #0];
      if ChInput = #0 then ChInput := ReadKey;
      DrawRect (Xp0, Yp0);
      IntersectLines (Xp0, Yp0);
      case ChInput of
         #75:  DEC(Xp0);    (* Left  arrow *)
         #77:  INC(Xp0);    (* Right arrow *)
         #72:  DEC(Yp0);    (* Up    arrow *)
         #80:  INC(Yp0)     (* Down arrow  *)
      end;
   until ChInput = #13;
   ZrMax.w := ANTILOG(Xval(Xp0));
{  ZrMax.w := Xval(Xp0);          }

   i := last;
   while (w_Hz^[i] > ZrMax.w) do DEC(i);
   Get_Xpt_Ypt (w_Hz^[i], Real_Z^[i]);
   Xp1 := Xpt; Yp1 := Ypt;
   Get_Xpt_Ypt (w_Hz^[i+1], Real_Z^[i+1]);
   Xp2 := Xpt; Yp2 := Ypt;
   repeat
      DrawRectLines (Xp0, Yp0, Xp1, Yp1, Xp2, Yp2);
      IntersectLines (Xp0, Yp0);
      repeat
         ChInput := ReadKey
      until ChInput in [#13, #0];
      if ChInput = #0 then ChInput := ReadKey;
      DrawRectLines (Xp0, Yp0, Xp1, Yp1, Xp2, Yp2);  (* Erase *)
      IntersectLines (Xp0, Yp0);                     (* Lines *)
      case ChInput of
         #75:  DEC(Xp0);    (* Left  arrow *)
         #77:  INC(Xp0);    (* Right arrow *)
         #72:  DEC(Yp0);    (* Up    arrow *)
         #80:  INC(Yp0)     (* Down arrow  *)
      end;
   until ChInput = #13;
   ZrMax.w  := AntiLOG(Xval(Xp0));
```

```
{  ZrMax.w  := Xval(Xp0);          }
   ZrMax.Zr := Yval(Yp0);

   ClearDevice;
   YLabel := 'Imaginary Z';
   GraphResults (w_Hz, Imag_Z, first, last, Lined, Rect);
   SetWriteMode (XORPut);
   ZrMax.Zi := Yo (ZrMax.w, w_Hz^[i], Imag_Z^[i],
                   w_Hz^[i+1], Imag_Z^[i+1]);
   Get_Xpt_Ypt (ZrMax.w, ZrMax.Zi);
   Xp0 := Xpt; Yp0 := Ypt;
   Get_Xpt_Ypt (w_Hz^[i], Imag_Z^[i]);
   Xp1 := Xpt; Yp1 := Ypt;
   Get_Xpt_Ypt (w_Hz^[i+1], Imag_Z^[i+1]);
   Xp2 := Xpt; Yp2 := Ypt;
   repeat
      DrawRectLines (Xp0, Yp0, Xp1, Yp1, Xp2, Yp2);
      IntersectLines (Xp0, Yp0);
      repeat
         ChInput := ReadKey
      until ChInput in [#13, #0];
      if ChInput = #0 then ChInput := ReadKey;
      DrawRectLines (Xp0, Yp0, Xp1, Yp1, Xp2, Yp2);  (* Erase *)
      IntersectLines (Xp0, Yp0);                     (* Lines *)
      case ChInput of
         #72:  DEC(Yp0);   (* Up   arrow *)
         #80:  INC(Yp0)    (* Down arrow *)
      end;
   until ChInput = #13;
   ZrMax.Zi := Yval (Yp0);
end;

procedure Insert_Zi0_ZrMax (var Hz, Real, Imag: DataPtr);
var
   i: integer;

   procedure MovePoints (var i: integer; Adder: integer;
                         Z: DataRec);
   begin
      while Hz^[i] > Z.w do begin
         Swap (Hz^[i+Adder], Hz^[i]);
         Swap (Real^[i+Adder], Real^[i]);
         Swap (Imag^[i+Adder], Imag^[i]);
         DEC(i)
         end;
      Hz^[i+Adder]   := Z.w;
```

```
      Real^[i+Adder] := Z.Zr;
      Imag^[i+Adder] := Z.Zi
   end;

begin
   i := NumPts;
   MovePoints (i, 2, ZrMax);
   MovePoints (i, 1, Zi0);
   NumPts := NumPts + 2;
end;

procedure AcceptInductive (var InductiveDone: Boolean);
var
   Accept: char;
begin
   if BeepON then Beeper;
   SmallHorizText;
   OutTextXY ((((CRTXmin+CRTXmax) DIV 2)-CRTXmin, CRTYmax DIV 2,
               'Accept points ?  ');
   repeat Read_KBD (Accept) until Accept in ['Y','N'];
   if Accept = 'Y'
      then InductiveDone := TRUE
      else InductiveDone := FALSE
end;

procedure Inductive_Data;
begin
   repeat
      Get_Zi0;
      Get_ZrMax;
      Nyquist_Plot (1, Numpts, Lined, Rect);
      AcceptInductive (InductiveDone)
   until InductiveDone;
   Insert_Zi0_ZrMax (w_Hz, Real_Z, Imag_Z);
end;

end.
```

B.8 Polynomial Fit of Experimental Data

```
unit DataFit;
{
   Subprograms for implementing polynomial fit of data set.
}
```

```
{$N+,E+}

interface

uses  Crt, Graph, KKDecl, GrafKK, Induct, TwoGraf, Gauss;


procedure Perform_Regression_and_Restructure_Frequency_Arrays
          (var Abort: boolean);

implementation

procedure High_frequency_asymptote;
var
   Dummy, LabelStr: string;
   StrWidth, X1, Y1, Yskip: integer;
   Xwidth, Ywidth: extended;
begin
   Xwidth := (CRTMaxX+1)/80;
   Ywidth := (CRTMaxY+1)/25;
   Title  := 'Determining High Frequency Asymptote';
   GraphResults (w_Hz, Real_Z, NumPts-10, NumPts, NotLined, Rect);
   SmallHorizText;
   STR (Real_Z^[NumPts]:12:5, Dummy);
   LabelStr :=  'Z'' at highest frequency              : ' + Dummy;
   OutTextXY (Xp(0.25), Yp(0.70), LabelStr);
   if BeepON then Beeper;
   LabelStr :=  'Enter high frequency asymptote for Z'': ';
   StrWidth := TextWidth (LabelStr);
   X1 := ROUND((Xp(0.25)+StrWidth)/Xwidth);
   Y1 := ROUND(Yp(0.75)/Ywidth);
   OutTextXY (Xp(0.25), ROUND((Y1+0.5)*Ywidth), LabelStr);
   { ReplyBox }
   Rectangle (Xp(0.25)+StrWidth, ROUND((Y1+1)*Ywidth)+2,
              Xp(0.40)+StrWidth, ROUND(Y1*Ywidth));
   case GraphDriver of
      VGA:  Yskip := 5;
     else   Yskip := 1
   end;
   gotoxy (X1+2, Y1+Yskip);
   read (Zr_oo)
end;

procedure Regression_of_Experimental_Data
          (var Y: DataPtr; var Z_fit: PolyFitPtr;
           var  Abort: Boolean);
```

```
var
   i, SegNum, First, Last: integer;
   Delta_x, Corr_Coef: extended;
   Accept: char;
begin
   SegNum := 0;
   repeat
      INC(SegNum);
      Corr_Coef := 0.0;
      FitOrder := 0;
      Get_Best_Fit (FitOrder, Index[SegNum-1], Index[SegNum],
                    Log_w, Y, Corr_Coef, Accept);
      Case Accept of
         'A': Abort := TRUE;
         'Y': if AutoRegress then
                 Store_PolyFit(ARFit.Order,ARFit.r,
                               ARFit.Coef,Z_fit^[SegNum])
              else
                 Store_PolyFit(FitOrder,Corr_Coef,
                               Poly_Coef,Z_fit^[SegNum])
      end;
   until (SegNum = Ne) or Abort
end;

procedure Calculate_Indices;       { for AutoIndex = TRUE }
var
   i: integer;
begin
   for i := 0 to (Ne-1) do
       Index[i] := 1 + ROUND(i*(NumPts/Ne));
   Index[Ne] := NumPts;
end;

procedure Perform_Regression_On_Experimental_Data
          (var Abort: boolean);
begin
   if AutoIndex then Calculate_Indices else Get_Indices;
   Title := 'Regression Fit --- Real Z';
   Plot_RealZ_Data (1, NumPts, NotLined, Rect);
   High_Frequency_Asymptote;
   Regression_of_Experimental_Data (Real_Z, Zr_fit, Abort);
   if Not Abort then begin
      Title := 'Regression Fit --- Imaginary Z';
      Plot_ImagZ_Data (1, NumPts, NotLined, Rect);
      Regression_of_Experimental_Data (Imag_Z, Zi_fit, Abort);
      if Not Abort then begin        { Regression performed }
```

```
         NewData := FALSE;
         Zfit_Hz := Hz_factor
         end;
      end;
   RestoreCRTMode;
end;

procedure Create_Expt_w_Arrays;
var
   i: integer;
begin
   for i := 1 to NumPts do begin
      w^[i] := w_Hz^[i] * Hz_factor;
      Log_w^[i] := LOG(w^[i])
      end;
end;

procedure Create_Low_w_Arrays;
var
   l: integer;
   Delta: extended;
begin
   w_0 :=  w_0_Hz * Hz_Factor;
   Delta := (Log(w_Hz^[1])-Log(w_0_Hz))/NewPts;
   for l := -NewPts to 0 do begin { Index for restored spectrum }
      Real_Z^[l] := 0.0;
      Imag_Z^[l] := 0.0;
      if l = 0
         then w_Hz^[0] := w_Hz^[1]
         else w_Hz^[l] := AntiLog(Log(w_0_Hz) + (l+NewPts)*Delta);
      w^[l] := w_Hz^[l] * Hz_factor;
      Log_w^[l] := LOG(w^[l]);
      end;

   { Integration limits for Adaptive Quad }
   Delta := (Log(w_Hz^[1])-Log(w_0_Hz))/Nl;
   for l := 0 to Nl-1 do
       Low_AB[l] := AntiLog(LOG(w_0_Hz) + l*Delta)*Hz_Factor;
   Low_AB[Nl] := w^[1];
end;

procedure Generate_Uniform_Frequency_Arrays;
begin
   ClrScr;
   Center (2,'LEAST SQUARE REGRESSION AND FREQUENCY ' +
             'ARRAY RESTRUCTURING');
```

```
   Center (3,'-------------------------------------' +
             '----------------------');
   if Changed_PolyOrder or Changed_w0 then begin
      if Changed_PolyOrder then
         WriteXY (10,5,'Polynomial order(s) has been changed ...');
      if Changed_w0 then
         WriteXY (10,6,'Lowest Frequency (w0) has been changed ...');
      if BeepON then Beeper;
         WriteXY(10, 7, 'Reindexing low frequency array ...');
      Create_Low_w_Arrays;
      Create_Expt_w_Arrays
      end;
end;

procedure Perform_Regression_and_Restructure_Frequency_Arrays
        (var Abort: Boolean);
begin
   if NewData then begin
      if Inductive AND (NOT InductiveDone) then Inductive_Data;
      Create_Expt_w_Arrays;
      Perform_Regression_on_Experimental_Data (Abort);
      end;
   if Not Abort then Generate_Uniform_Frequency_Arrays;
end;

end.
```

B.9 Gaussian Elimination Routine

```
unit Gauss;
{
   Performs Gaussian elimination routine with scaled-column
   pivoting ... see R.L. Burden and J.D. Faires,
   ''Numerical Analysis,'' (4nd ed.), Prindle, Weber
   & Schmidt, Boston, MA, p.332, (1989).
}

{$N+,E+}

interface

uses  Crt, Graph, KKDecl, GrafKK;

procedure GaussPtrsToNil;
procedure AllocateGaussArrays;
```

```
procedure DeallocateGaussArrays;
procedure Initialize_Variables;
procedure Gaussian_Elimination (N: integer);
procedure Get_Best_Fit (var FitOrder: integer; First, Last: integer;
                        var X, Y: DataPtr;  var Corr_Coef: extended;
                        var Accept: char);
procedure Store_PolyFit (FitOrder: integer; r: extended;
                         Poly_Coef: Vector; var Z_fit: PolyFitRecord);

implementation

var
   AbortGaussian : boolean;
   Sum_Y         : extended;

procedure GaussPtrsToNil;
begin
   A := nil;
   B := nil;
   S := nil;
   Sum_Xn := nil;
   Sum_YXn := nil
end;

procedure AllocateGaussArrays;
begin
   GaussPtrsToNil;
   new (A);
   new (B);
   new (S);
   new (Sum_Xn);
   new (Sum_YXn)
end;

procedure DeallocateGaussArrays;
begin
   dispose (A);
   dispose (B);
   dispose (S);
   dispose (Sum_Xn);
   dispose (Sum_YXn)
end;

procedure Initialize_Variables;
var
   i, j: integer;
```

```
begin
   for i := 0 to Max_Poly_order do begin
      Poly_Coef[i] := 0.0;
      B^[i] := 0.0;
      for j := 0 to Max_Poly_order do  A^[i,j] := 0.0;
      end;
end;

procedure NoSolution;
begin
   AbortGaussian := True;
   writeln ('NO Unique solution exists')
end;

procedure Step_1 (N: integer);
var
   i, j: integer;
   Max: extended;
begin
   for i := 0 to N do begin
       Max := ABS(A^[i, 0]);
       for j := 1 to N do
           if ABS(A^[i,j]) > Max then Max := ABS(A^[i,j]);
       S^[i] := Max;
       if S^[i] = 0 then NoSolution;
       Nrow[i] := i
       end
end;

procedure Step_3 (N, i: integer; var p: integer);
var
   j: integer;
   Max: extended;
begin
   p := i;
   Max := ABS(A^[Nrow[p], i])/S^[Nrow[p]];
   for j := i+1 to N do
       if ABS(A^[Nrow[j], i])/S^[Nrow[j]] > Max then begin
          Max := ABS(A^[Nrow[j], i])/S^[Nrow[j]];
          p := j
          end
end;

procedure Steps_6_to_9 (i, N: integer);
var
   j, k: integer;
```

```
begin
   for j := i+1 to N do begin
      A^[Nrow[j],i] := A^[Nrow[j],i] / A^[Nrow[i],i];   { Step 7 }
      for k := i+1 to N do                               { Step 8a }
          A^[Nrow[j],k] := A^[Nrow[j],k] -
                           A^[Nrow[j],i]*A^[Nrow[i],k];
      B^[Nrow[j]] :=  B^[Nrow[j]] -
                      A^[Nrow[j],i]*B^[Nrow[i]]          { Step 8b }
      end;
   if A^[Nrow[N],N] = 0 then NoSolution                  { Step 9 }
end;

procedure Steps_2_to_9 (N: integer);
var
   i, p: integer;
begin
   for i := 0 to N-1 do begin       { Step 2 }
       Step_3 (N, i, p);
       if A^[Nrow[p], i] = 0 then       { Step 4 }
          NoSolution
       else begin
          if Nrow[i] <> Nrow[p] then    { Step 5: Simulated
                                                  Row Exchange }
          SwapINT (Nrow[i], Nrow[p]);
          Steps_6_to_9 (i, N)           { Reduce matrix elements }
          end
       end
end;

procedure Steps_10_to_12 (N: integer);
var
   i, j: integer;
   Sum: extended;
begin
   Poly_Coef[N] := B^[Nrow[N]]/A^[Nrow[N],N];
   for i := N-1 downto 0 do begin
       Sum := 0.0;
       for j := i+1 to N do
           Sum := Sum + A^[Nrow[i],j]*Poly_Coef[j];
       Poly_Coef[i] := (B^[Nrow[i]] - Sum)/A^[Nrow[i],i]
       end
end;

procedure Gaussian_Elimination (N: integer);
begin
   AbortGaussian := False;
```

```
   Step_1 (N);
   if NOT AbortGaussian then Steps_2_to_9 (N);
   if NOT AbortGaussian then Steps_10_to_12 (N);
end;

procedure Enter_LHS_of_Normal_Eqn (N, First, Last: integer;
                                   var X, Y: DataPtr);
var
   i, j: integer;
begin
   for j := 0 to Max_poly_order do   { Initialize LHS vector }
       Sum_YXn^[j] := 0;
   for i := First to Last do         { Enter LHS vector }
       for j := 0 to N do
           Sum_YXn^[j] := Sum_YXn^[j] + Y^[i]*exp(j*ln(X^[i]));
   Sum_Y := Sum_YXn^[0];             { Assign dY }
   for i := 0 to N do                { Assign B-vector }
       B^[i] := Sum_YXn^[N-i];
end;

procedure Enter_Normal_Eqn_Matrix (N, First, Last: integer);
var
   i, j: integer;
begin
   for i := 0 to N do
      for j := 0 to N do
          if (i=N) and (j=0)
             then A^[i,j] := Last-First+1
             else A^[i,j] := Sum_Xn^[N+j-i];
end;

procedure Enter_Sum_X_Matrix (N, First, Last: integer; var X: DataPtr);
var
   i, j: integer;
begin
   for j := 1 to Twice_Max_poly_order do   { Initialize Sum_X vector }
       Sum_Xn^[j] := 0;
   for i := First to Last do               { Enter Sum_X vector }
       for j := 1 to (2*N) do
           Sum_Xn^[j] := Sum_Xn^[j] + exp(j*ln(X^[i]));
   Enter_Normal_Eqn_Matrix (N, First, Last);
end;

procedure Estimate_Corr_Coef (N, First, Last: integer;
                              var X, Y, Y_ic: DataPtr;
                              var r: extended);
```

```
var
   Y_mean, Sum, S_y_squared, S_yx_squared: extended;
   i, j, TotalPts , No_of_parameters: integer;
begin
   No_of_parameters := N + 1;
   TotalPts := Last - First + 1;      { No of Points in the segment }
   Y_mean := Sum_Y/TotalPts;          { Mean of Y }
{  Standard Deviation of Y-axis data, S_y  }
   Sum := 0;
   for i := First to Last do
       Sum := Sum + (Y^[i] - Y_mean)*(Y^[i] - Y_mean);
   S_y_squared :=  Sum/(TotalPts - 1);
{  Standard Error of Estimate, S_yx   }
   Sum := 0;
   for i := First to Last do begin
       Y_ic^[i] := Poly_Y (N, Poly_Coef, X^[i]);
       Sum := Sum + (Y^[i] - Y_ic^[i])*(Y^[i] - Y_ic^[i]);
       end;
   S_yx_squared :=  Sum/(TotalPts - No_of_parameters);
{  Correlation Coefficient, r  }
   if S_y_squared = 0.0
      then r := 1.00
      else r := SQRT(abs(1 - S_yx_squared/S_y_squared));
end;

procedure Poly_Fit (var N, First, Last: integer; var X,Y: DataPtr);
begin
   AllocateGaussArrays;
   Initialize_variables;
   Enter_LHS_of_Normal_Eqn (N, First, Last, X, Y);
   Enter_Sum_X_Matrix (N, First, Last, X);
   Gaussian_Elimination (N);
   DeallocateGaussArrays;
end;

procedure Plot_Segment (first, last: integer; var X, Y: DataPtr);
begin
   Title := '';
   GraphResults (X, Y, first, last, NotLined, Rect);
   SmallHorizText;
   Xpt := Xp(0.722); Ypt := Yp(0.133);
   DrawFigure (Rect);
   OutTextXY (Xp(0.75), Ypt, 'Expt''l data');
   OutTextXY (Xp(0.7125), Yp(0.167), '-- Best-fit');
   OutTextXY (Xp(0.025), Yp(0.94), '<ESC> to Abort')
end;
```

```
procedure Write_Fit_Results (r: double; FitOrder: integer);
var
   Dummy: String;
begin
   SetViewPort (CRTXmin, 1, CRTXmax, CRTYmax-1, ClipON);
   ClearViewPort;
   SmallHorizText;
   STR (r:7:4, Dummy);
   OutTextXY (10, (CRTYmax DIV 3), 'r = ' + Dummy);
   STR (FitOrder:2, Dummy);
   OutTextXY (10, ((CRTYmax*2) DIV 3), 'Poly Order = ' + Dummy);
   if NOT AutoRegress then
   OutTextXY (((CRTXmin+CRTXmax) DIV 2)-CRTXmin, CRTYmax DIV 2,
              'Accept best-fit (Y/N)?  ');
   SetViewPort (0, 0, CRTMaxX, CrtMaxY, ClipON)
end;

procedure Store_PolyFit (FitOrder: integer;  r: extended;
                         Poly_Coef: Vector;
                         var Z_fit: PolyFitRecord);
var
   j: integer;
begin
   Z_fit.Order := FitOrder;
   Z_fit.r := r;
   for j := 0 to FitOrder do
       Z_fit.Coef[j] := Poly_Coef[j];
end;

procedure AutoDelay;
begin
   if FastAuto then Delay (250) else Delay (1500)
end;

procedure AcceptFit (var FirstTime: Boolean;  r: extended;
                     var FitOrder: integer; MaxOrder: integer;
                     var Accept: char);
begin
   if AutoRegress then begin
      if FirstTime then begin
         Store_PolyFit (FitOrder, r, Poly_Coef, ARFit);
         FirstTime := FALSE
         end;
      if ((r<CorrCoefOK) or (r>1.0)) and (FitOrder<Max_Poly_Order)
         then inc(FitOrder)
```

```
         else Accept := 'Y';
      if (FitOrder = MaxOrder) and (Accept = 'N') then begin
         dec (FitOrder);
         Accept := 'Y'
         end;
      if ((r >= ARFit.r) and (r <= 1.0)) or (Accept = 'Y') then
         Store_PolyFit (FitOrder, r, Poly_Coef, ARFit);
      AutoDelay;
      if keypressed AND (ReadKey = #27) then Accept := 'A'
      end
   else begin   { NOT AutoRegress }
      if BeepON then Beeper;
      repeat Read_KBD (Accept) until Accept in ['Y','N','A'];
      if Accept = 'N' then begin
         if (FitOrder<MaxOrder-1) AND (FitOrder<Max_Poly_Order)
            then inc (FitOrder)
            else Accept := 'Y';
          end
      end
end;

procedure Straight_Line (var X,Y:DataPtr; first,last:integer;
                         var fitorder:integer; var r:extended;
                         var Accept: char);
begin
   Poly_coef[1] := (Y^[last]-Y^[first])/(X^[last]-X^[first]);
   Poly_coef[0] :=  (Y^[first] - Poly_coef[1]*X^[first]);
   r := 1.0;
   FitOrder := 1;
   Accept := 'Y';
   GraphResults (X, Y, first, last, Lined, Rect);
   GraphPause (Xp(0.25), Yp(0.75), 2);
   if AutoRegress then begin
      Store_PolyFit (FitOrder, r, Poly_Coef, ARFit);
      AutoDelay
      end
end;

procedure Get_Best_Fit (var FitOrder: integer; First, Last: integer;
                        var X, Y: DataPtr; var Corr_Coef: extended;
                        var Accept: char);
var
   Y_ic: DataPtr;
   i: integer;
   FirstTime: Boolean;
begin
```

```
   Y_ic := nil;  new (Y_ic);
   FirstTime := TRUE;  Accept := 'N';
   if (Last-First+1) = 2 then
      Straight_Line (X, Y, First, Last, FitOrder, Corr_Coef, Accept)
   else begin
      Plot_Segment (first, last, w_Hz, Y);
      repeat
         Poly_Fit (FitOrder, First, Last, X, Y);
         Estimate_Corr_Coef (FitOrder, First, Last,
                             X, Y, Y_ic, Corr_Coef);
         Write_Fit_Results (Corr_Coef, FitOrder);
         SetWriteMode (XORPut);
         PlotXY (w_Hz, Y_ic, first, last, Lined, NoSymbol);
         AcceptFit (FirstTime, Corr_Coef, FitOrder,
                    Last-First, Accept);
         if Accept = 'N' then       { Erase labels }
         PlotXY (w_Hz, Y_ic, first, last, Lined, NoSymbol);
         SetWriteMode (CopyPut)
      until (Accept <> 'N') or (FitOrder > Last-First)
      end;
   dispose (Y_ic);
end;

end.
```

B.10 Slopes at Truncation Frequency

```
unit GetSlope;
{   Calculate slope at truncation frequency using 5-pt formula  }

{$N+,E+}

interface

uses  Crt, KKDecl;

procedure Calculate_Slopes;

implementation

function FivePointSlope (var X, Y: DataPtr; Xo, h: integer): extended;
var
   StepSize: extended;
begin
   StepSize := X^[Xo] - X^[Xo+h];
```

```
   FivePointSlope := (-25*Y^[Xo] + 48*Y^[Xo+h] - 36*Y^[Xo+2*h]
                     + 16*Y^[Xo+3*h] - 3*Y^[Xo+4*h])/(12*StepSize)
end;

procedure Calculate_Slopes;
var
   w_trunc, Real_Left, Real_Right,
   Imag_Left, Imag_Right: extended;
   i, j, Xo_L, Xo_R: integer;
   Log_w_Hz: DataPtr;
begin
   Log_w_Hz := nil; new (Log_w_Hz);
   ClrScr;
   Center (2, 'Calculating Slope At Truncation Frequency ' +
              '(5-Pt Method)');
   Center (3, '-------------------------------------------' +
              '---------------------');
   WriteXY (5, 5, ' Frequency, Hz         Log (w)  '+
                  '          Real, Z''          Imag, Z"');
   writeln;
   for j := -6 to +6 do begin
       Log_w_Hz^[j] := LOG (w_Hz^[j]);
       writeln (w_Hz^[j]:13:5, Log_w_Hz^[j]:19:5,
                Real_Z^[j]:20:4, Imag_Z^[j]:20:5)
       end;
   Real_Left  := FivePointSlope (Log_w_Hz, Real_Z, 0, -1);
   Real_Right := FivePointSlope (Log_w_Hz, Real_Z, 1, 1);
   Imag_Left  := FivePointSlope (Log_w_Hz, Imag_Z, 0, -1);
   Imag_Right := FivePointSlope (Log_w_Hz, Imag_Z, 1, 1);
   writeln; writeln;
   WriteXY (10,5, 'Real Impedance: (Ohms/decade)'); writeln;
   writeln ('                  Calculated   Region = ',
            Real_Left:15:5);
   writeln ('                  Experimental Region = ',
            Real_Right:15:5);
   WriteXY (10,9, 'Imag Impedance: (Ohms/decade)'); writeln;
   writeln ('                  Calculated   Region = ',
            Imag_Left:15:5);
   writeln ('                  Experimental Region = ',
            Imag_Right:15:5);
   CRKey ('Press <Enter> to continue ...');
   dispose (Log_w_Hz)
end;

end.
```

B.11 Graphics Routine for Data Deletion

```
unit TwoGraf;
{
   Subprogram contains plotting instructions for data deletion
}

{$N+,E+}

interface

uses  Graph, Crt, KKDecl, GrafKK;

procedure DeletePoints;
procedure Get_Indices;

implementation

const  RealZ         = 1;
       ImagZ         = 2;

var    VID_Xmin,  VID_Xmax,
       VID_Y1min, VID_Y1max,
       VID_Y2min, VID_Y2max,
       VID_VMajTicLen, VID_VMinTicLen,
       VID_HMajTicLen, VID_HMinTicLen,
       DelPts                          : integer;

       DelIndex:  array [1..(MaxDataPts DIV 2)] of integer;
       DeletePt:  array [1..MaxDataPts] of Boolean;

       Xlabel, Y1label, Y2label: str40;
       Title: Str80;

       Y1min, Y1max, Y1_tic_value,
       Y2min, Y2max, Y2_tic_value    : extended;

function Xp1 (X: double): integer;
begin
  Xp1 := ROUND(VID_Xmin + (X - Xmin)/(Xmax - Xmin)*
              (VID_Xmax - VID_Xmin))
end;

function Yp1 (Y: double): integer;
begin
   Yp1 := ROUND(VID_Y1min + (Y - Y1min)/(Y1max - Y1min)*
```

```
                (VID_Y1max - VID_Y1min))
end;

function Yp2 (Y: double): integer;
begin
   Yp2 := ROUND(VID_Y2min + (Y - Y2min)/(Y2max - Y2min)*
                (VID_Y2max - VID_Y2min))
end;

procedure Xpt_Ypt (X, Y: double; WhichGraph: integer);
begin
   Xpt := Xp1 (X);
   case WhichGraph of
      RealZ :  Ypt := Yp1(Y);
      ImagZ :  Ypt := Yp2(Y)
   end
end;

procedure XMajorTics (X: double; Decimal: boolean;
                      WhichGraph: integer);
var
   XticLabel  : LongInt;
   MajTicLabel: string[10];
begin
   Xpt_Ypt (X, Ymin, WhichGraph);
   Line (Xpt, Ypt, Xpt, Ypt - VID_VMajTicLen);
   Xpt_Ypt (X, Ymax, WhichGraph);
   Line (Xpt, Ypt, Xpt, Ypt + VID_VMajTicLen);
   if Decimal then
      STR (X:7:2, MajTicLabel)
   else begin
      XticLabel := ROUND(X);
      STR (XticLabel, MajTicLabel)
      end;
   if LogXaxis then begin
      SetTextJustify(CenterText, CenterText);
      SetTextStyle (SmallFont, HorizDir, 0); { LogSize);}
      OutTextXY(Xpt, VID_Y2min + 15, '10');
      SetTextJustify (LeftText, CenterText);
      SetTextStyle (SmallFont, HorizDir, 0); { LogPow); }
      OutTextXY(Xpt + 11, VID_Y2min + 10, MajTicLabel)
      end
   else
      OutTextXY(Xpt, VID_Y2min + 10, MajTicLabel)
end;
```

```
procedure XMinorTics (X: double; WhichGraph: integer);
begin
   case WhichGraph of
      RealZ:  begin  Ymin := Y1min; Ymax := Y1max  end;
      ImagZ:  begin  Ymin := Y2min; Ymax := Y2max  end
   end;
   Xpt_Ypt (X, Ymin, WhichGraph);
   Line (Xpt, Ypt, Xpt, Ypt - VID_VMinTicLen);
   Xpt_Ypt (X, Ymax, WhichGraph);
   Line (Xpt, Ypt, Xpt, Ypt + VID_VMinTicLen);
end;

procedure YMajorTics (Y: double; Decimal: Boolean;
                      WhichGraph: integer);
var
   YticLabel  : LongInt;
   MajTicLabel: string[10];
begin
   Xpt_Ypt (Xmin, Y, WhichGraph);
   Line (Xpt, Ypt, Xpt + VID_HMajTicLen, Ypt);
   Xpt_Ypt (Xmax, Y, WhichGraph);
   Line (Xpt, Ypt, Xpt - VID_HMajTicLen, Ypt);
   if Decimal then
      STR (Y:10:2, MajTicLabel)
   else begin
      YticLabel := ROUND(Y);
      STR (YticLabel, MajTicLabel)
      end;
   if LogYaxis then begin
      SetTextJustify(RightText, CenterText);
      SetTextStyle (SmallFont, HorizDir, 0); { LogSize); }
      OutTextXY(VID_XMin - 20, Ypt, '10');
      SetTextJustify (LeftText, CenterText);
      SetTextStyle (SmallFont, HorizDir, 0); { LogPow); }
      OutTextXY(VID_XMin - 18, Ypt - 9, MajTicLabel)
      end
   else
      OutTextXY(VID_XMin - 10, Ypt, MajTicLabel)
end;

procedure YMinorTics (Y: double; WhichGraph: integer);
begin
   Xpt_Ypt (Xmin, Y, WhichGraph);
   Line (Xpt, Ypt, Xpt + VID_HMinTicLen, Ypt);
   Xpt_Ypt (Xmax, Y, WhichGraph);
   Line (Xpt, Ypt, Xpt - VID_HMinTicLen, Ypt);
```

```
end;

procedure LinearTics (Min, Max, Tic_value: extended;
                      Tic_space, Axis, WhichGraph:integer);
var
   I: extended;
   J, MajorIntervals, NumLoops: integer;
   MajorTic, Decimal: Boolean;
begin
   case WhichGraph of
      RealZ:  begin  Ymin := Y1min; Ymax := Y1max  end;
      ImagZ:  begin  Ymin := Y2min; Ymax := Y2max  end
   end;
   I := Min;
   NumLoops := ROUND((Max-Min)/(Tic_value * Tic_space));
   if Tic_Value*Tic_Space < 5.0 then
      Decimal := TRUE
   else
      Decimal := False;
   for MajorIntervals := 1 to NumLoops do begin
       MajorTic := TRUE;
       for J := 1 to Tic_Space do begin
           if MajorTic then begin
              case Axis of
                 Xaxis:  XMajorTics (I, Decimal, WhichGraph);
                 Yaxis:  YMajorTics (I, Decimal, WhichGraph)
              end;
              MajorTic := FALSE
              end
           else
              case Axis of
                 Xaxis:  XMinorTics (I, WhichGraph);
                 Yaxis:  YMinorTics (I, WhichGraph)
              end;
           I := I + Tic_Value
           end
       end;
       case Axis of         { Maximum Tic mark and Label }
           Xaxis:  XMajorTics (Max, Decimal, WhichGraph);
           Yaxis:  YMajorTics (Max, Decimal, WhichGraph)
       end;
end;

procedure LogTics (LogMin, LogMax, Axis, WhichGraph: integer);
var
   LogI, Loop: integer;
```

```
   Low, High, J, LogJ, Delta: extended;
   MajorTic: Boolean;
begin
   case WhichGraph of
      RealZ:  begin  Ymin := Y1min; Ymax := Y1max  end;
      ImagZ:  begin  Ymin := Y2min; Ymax := Y2max  end
   end;
   for LogI := LogMin to (LogMax - 1) do begin
       Low  := AntiLog (LogI);   J := Low;
       High := AntiLog (LogI+1); Delta := High/10.;
       MajorTic := TRUE;  Loop := 0;
       repeat
          Loop := Loop + 1;
          LogJ := LOG(J);
          if MajorTic then  begin
             case Axis of
                Xaxis:  XMajorTics (LogJ, False, WhichGraph);
                Yaxis:  YMajorTics (LogJ, False, WhichGraph)
             end;
             MajorTic := FALSE
             end
          else begin
             case Axis of
                Xaxis:  XMinorTics (LogJ, WhichGraph);
                Yaxis:  YMinorTics (LogJ, WhichGraph)
             end
             end;
          J := J + Delta;
       until Loop = 9
       end;
       case Axis of
          Xaxis:  XMajorTics (LogMax, False, WhichGraph);
          Yaxis:  YMajorTics (LogMax, False, WhichGraph)
       end
end;

procedure DrawTicMarks (var Ymin, Ymax, Y_tic_value: extended;
                        WhichGraph: integer);
begin
   if LogXaxis then       (*  Draw X-Axis tic marks  *)
      LogTics (TRUNC(Xmin), TRUNC(Xmax), Xaxis, WhichGraph)
   else  begin
      SetTextJustify(CenterText, TopText);
      SetTextStyle (SmallFont, Horizdir, 0); {SmallFontSize);}
      LinearTics (Xmin, Xmax, X_tic_value,
                  X_tic_space, Xaxis, WhichGraph)
```

```
      end;
   if LogYaxis then       (*  Draw Y-axis tic marks  *)
      LogTics (TRUNC(Ymin), TRUNC(Ymax), Yaxis, WhichGraph)
   else  begin
      SetTextJustify (RightText, CenterText);
      SetTextStyle (SmallFont, HorizDir, 0); { SmallFontSize);}
      LinearTics (Ymin, Ymax, Y_tic_value,
                  Y_tic_space, Yaxis, WhichGraph)
      end
end;

procedure LabelGraph;
var
   X1, Y1, Y2, X3, Y3: integer;
begin
   SetTextJustify (CenterText, CenterText);
   SetTextStyle (SmallFont, HorizDir, 6); {LabelSize);}
   X1 := (VID_Xmax + VID_Xmin) DIV 2;
   Y1 := VID_Y1max DIV 2;
   OutTextXY (X1, Y1, Title);

   Y2 := VID_Y2min + (2*(CRTMaxY - VID_Y2min) DIV 3);
   OutTextXY (X1, Y2, XLabel);

   SetTextStyle (SmallFont, VertDir, 6); {LabelSize);}
   X3 := VID_Xmin DIV 4;
   Y3 := (VID_Y1Max + VID_Y1min) DIV 2;
   OutTextXY (X3, Y3, Y1Label);

   Y3 := (VID_Y2Max + VID_Y2min) DIV 2;
   OutTextXY (X3, Y3, Y2Label)
end;

procedure Get_Xpt_Ypt (X, Y: double; WhichGraph: integer);
begin
   if LogXaxis then begin
      if LogYaxis
         then Xpt_Ypt (LOG(X), LOG(Y), WhichGraph)
         else Xpt_Ypt (LOG(X), Y, WhichGraph)
      end
   else begin
      if LogYaxis
         then Xpt_Ypt (X, LOG(Y), WhichGraph)
         else Xpt_Ypt (X, (Y), WhichGraph)
      end
end;
```

```
procedure PlotData (var X, Y: DataPtr;
                    First, Last, WhichGraph, Liner: integer;
                    FigType: Figures);
var
   i: integer;
   X1, Y1: integer;
   TempFig: Figures;
begin
   for i := first to last do
       begin
       Get_Xpt_Ypt (X^[i], Y^[i], WhichGraph);
       if FigType <> NoSymbol then begin
          if (i<=0)
             then TempFig := Triangle
             else TempFig := FigType;
          DrawFigure (TempFig)
          end;
       if Liner = Lined then begin
          if i > first then Line (X1, Y1, Xpt, Ypt);
          X1 := Xpt;
          Y1 := Ypt
          end
       end
end;

procedure Draw_Two_Graphs;
begin
   SetGraphMode(GraphMode);
   ClearDevice;
   if MaxColor > 1 then begin
      SetBkColor(Blue);
      SetColor(Yellow)
      end;
   SetViewPort (0, 0, CRTMaxX, CrtMaxY, ClipON);
   Rectangle (0, 0, CRTMaxX, CrtMaxY);

   (*  DRAW REAL DATA  *)
   Rectangle (VID_Xmin, VID_Y1min, VID_Xmax, VID_Y1max);
   Auto_Plot_Parameters (w_Hz, Real_Z, 1, NumPts);
   Y1min := Ymin;  Y1max := Ymax; Y1_tic_value := Y_tic_value;
   DrawTicMarks (Y1min, Y1max, Y1_tic_value, RealZ);
   PlotData (w_Hz, Real_Z, 1, NumPts, RealZ, NotLined, Rect);

   (*  DRAW IMAGINARY DATA *)
   Rectangle (VID_Xmin, VID_Y2min, VID_Xmax, VID_Y2max);
```

```
   Auto_Plot_Parameters (w_Hz, Imag_Z, 1, NumPts);
   Y2min := Ymin;  Y2max := Ymax; Y2_tic_value := Y_tic_value;
   DrawTicMarks (Y2min, Y2max, Y2_tic_value, ImagZ);
   PlotData (w_Hz, Imag_Z, 1, NumPts, ImagZ, NotLined, Rect);
end;

procedure Get_TwoGraph_Parameters;
begin
   VID_Xmin  := Xp(0.20);   VID_Xmax  := Xp(0.95);
   VID_Y1min := Yp(0.45);   VID_Y1max := Yp(0.05);
   VID_Y2min := Yp(0.90);   VID_Y2max := Yp(0.50);

   VID_VMajTicLen := Yp(0.03);
   VID_VMinTicLen := VID_VMajTicLen DIV 2;
   VID_HMajTicLen := Xp(0.015);
   VID_HMinTicLen := VID_HMajTicLen DIV 2;

   Xlabel  := 'Frequency, Hz';
   Y1label := 'Real Impedance';
   Y2label := 'Imaginary Impedance';
   LogXaxis := TRUE;
   LogYaxis := FALSE
end;

procedure DrawCross (X, Y: extended; WhichGraph: integer);
begin
   Get_Xpt_Ypt (X, Y, WhichGraph);
   Line (Xpt-4,Ypt,Xpt+4,Ypt);
   Line (Xpt,Ypt-4,Xpt,Ypt+4)
end;

procedure DrawCrosses (i: integer);
begin
   DrawCross (w_Hz^[i], Real_Z^[i], RealZ);   { Draw Cross }
   DrawCross (w_Hz^[i], Imag_Z^[i], ImagZ);
end;

procedure Get_Indices;              { for AutoIndex = FALSE }
var
   i: integer;
   KeyInput: integer;
   Abort: boolean;
begin
   Get_TwoGraph_Parameters;
   Draw_Two_Graphs;
   Title := 'Segment EndPoints: <Enter> to select... <ESC> to end';
```

```
   LabelGraph;
   SetWriteMode (XORPut);
   Abort := FALSE;
   Index[0] := 1;
   i := 2; Ne := 0;
   if MaxColor>1 then SetColor(White);
   while NOT Abort do begin
      DrawCrosses (i);   { Draw Cross }
      KeyInput := ScanKey;
      DrawCrosses (i);   { Erase Cross }
      case KeyInput of
           CR: begin
                  if i <= Index[Ne] then begin
                     if BeepON then Beeper
                     end
                  else begin
                     INC(Ne);
                     if (Ne < MaxSegments) then
                        Index[Ne] := i
                     else begin
                        Index[Ne] := NumPts;
                        Abort := TRUE
                        end;
                     Get_Xpt_Ypt (w_Hz^[i], Real_Z^[i], RealZ);
                     DrawFigure (Circ);
                     Get_Xpt_Ypt (w_Hz^[i], Imag_Z^[i], ImagZ);
                     DrawFigure (Circ);
                     if (i=NumPts) then Abort := TRUE
                     end
               end;
          ESC: begin
                  Abort := TRUE;
                  INC(Ne);
                  Index[Ne] := NumPts
               end;
         LEFT: if i>1 then DEC(i);          { Left  arrow }
        RIGHT: if i<NumPts then INC(i);     { Right arrow }
         else  if BeepON then Beeper
      end  { case }
      end  { while }
end;

procedure MarkPoint (i: integer);

   procedure ShowCircle;
   begin
```

```
      DrawFigure (Circ);
   end;

   procedure HideCircle;
   var
      TempColor: Word;
   begin
      TempColor := GetColor;
      SetColor (GetBkColor);
      ShowCircle;
      SetColor (TempColor)
   end;

begin
   Get_Xpt_Ypt (w_Hz^[i], Real_Z^[i], RealZ);
   if DeletePT[i] then ShowCircle else HideCircle;
   Get_Xpt_Ypt (w_Hz^[i], Imag_Z^[i], ImagZ);
   if DeletePT[i] then ShowCircle else HideCircle
end;

procedure Choose_Points;
var
   i, j: integer;
   KeyInput: integer;
   Abort: boolean;
begin
   SetWriteMode (XORPut);
   Abort := FALSE;
   for i := 1 to MaxDataPts do
       DeletePT[i] := FALSE;
   i := 1;
   if MaxColor>1 then SetColor(White);
   while NOT Abort do begin
        DrawCrosses (i);   { Draw Cross }
        KeyInput := ScanKey;
        DrawCrosses (i);   { Erase Cross }
        case KeyInput of
              CR:  begin
                     DeletePT[i] := NOT DeletePT[i];
                     MarkPoint (i);
                   end;
             ESC:  Abort := TRUE;
            HOME:  i := 1;
          ENDKey:  i := NumPts;
            LEFT:  if i>1 then DEC(i);        { Left  arrow }
           RIGHT:  if i<NumPts then INC(i);   { Right arrow }
```

```
            else   if BeepON then Beeper
         end  { case }
      end;   { while }
   for i := 1 to  (MaxDataPts Div 2) do
       DelIndex[i] := 0;
   DelPts := 0;
   for i := 1 to NumPts do     { Assign DelIndex array }
       if DeletePt[i] then begin
          INC(DelPts);
          DelIndex[DelPts] := i
          end
end;

procedure Sort_DelIndex_Array (Low, High: integer);
{
   QUICKSORT Algorithm adapted from "Complete Turbo Pascal",
   3rd ed, by J. Duntemann, (C) 1989
}
var
   i, j, Pivot: integer;
begin
   if Low < High then begin
      i := Low;
      j := High;
      Pivot := DelIndex[j];
      repeat
         while (i<j) and (DelIndex[i] <= Pivot) do INC (i);
         while (j>i) and (DelIndex[j] >= Pivot) do DEC (j);
         if i < j then SwapINT (DelIndex[i], DelIndex[j])
      until i >= j;
      SwapInt (DelIndex[i], DelIndex[High]);
      if (i-Low < High -i) then begin
         Sort_DelIndex_Array (Low, i-1);
         Sort_DelIndex_Array (i+1, High)
         end
      else begin
         Sort_DelIndex_Array (i+1, High);
         Sort_DelIndex_Array (Low, i-1)
         end
      end
end;

procedure Delete_Points_Chosen;
var
   i, j: integer;
begin
```

```
   i := 1;
   j := DelIndex[i] + 1;
   repeat
      if (i < DelPts) and (j = DelIndex[i+1]) then
         inc(i)
      else begin
         w_Hz^[j-i]   := w_Hz^[j];
         Real_Z^[j-i] := Real_Z^[j];
         Imag_Z^[j-i] := Imag_Z^[j]
         end;
      inc(j)
   until j > NumPts;
   NumPts := NumPts - DelPts;
end;

procedure DeletePoints;
var
   DelPtStr: string;
begin
   Get_TwoGraph_Parameters;
   Draw_Two_Graphs;
   Title := 'Delete Points:  <Enter> to select ... <ESC> to end';
   LabelGraph;
   Choose_Points;
   if DelPts <> 0 then begin
      Delete_Points_Chosen;
      NewData := TRUE;  HaveParameters := FALSE;
      Changed_PolyOrder := TRUE; Changed_w0 := TRUE;
      end;
   Draw_Two_Graphs;
   if DelPts = 0 then
      Title := 'No Points were Deleted:  <CR> to continue'
   else  begin
      Str (DelPts, DelPtStr);
      Title := DelPtStr + ' Points Deleted:  <Enter> to continue'
      end;
   LabelGraph;
   CRKey ('');
   RestoreCRTMode
end;

end.
```

B.12 Graphics Fonts

```
unit Fonts;
{
  This unit links all the BGI graphics fonts into a
  single TPU file.  This makes it easy to incorporate the
  font files directly into an .EXE file.  This sample unit
  can found in the Borland Turbo Pascal examples library.
  Copyright (c) 1985, 1989 by Borland International, Inc.
}

interface

procedure GothicFontProc;
procedure SansSerifFontProc;
procedure SmallFontProc;
procedure TriplexFontProc;

implementation

procedure GothicFontProc; external;
{$L GOTH.OBJ }

procedure SansSerifFontProc; external;
{$L SANS.OBJ }

procedure SmallFontProc; external;
{$L LITT.OBJ }

procedure TriplexFontProc; external;
{$L TRIP.OBJ }

end.
```

B.13 Video Display Drivers

```
unit Drivers;
{
  This unit links the BGI graphics driver into a single
  TPU file. This makes it easy to link the driver files
  directly into an .EXE file.  This sample unit can
  found in the Borland Turbo Pascal examples library.
  Copyright (c) 1985, 1989 by Borland International, Inc.
}
```

```
interface

procedure ATTDriverProc;
procedure CgaDriverProc;
procedure EgaVgaDriverProc;
procedure HercDriverProc;
procedure PC3270DriverProc;
procedure IBM8514DriverProc;

implementation

procedure ATTDriverProc; external;
{$L ATT.OBJ }

procedure CgaDriverProc; external;
{$L CGA.OBJ }

procedure EgaVgaDriverProc; external;
{$L EGAVGA.OBJ }

procedure HercDriverProc; external;
{$L HERC.OBJ }

procedure PC3270DriverProc; external;
{$L PC3270.OBJ }

procedure IBM8514DriverProc; external;
{$L IBM8514.OBJ }

end.
```

Appendix C
POTENTIODYNAMIC SCANS FOR RRDE

Potentiodynamic scans for an illuminated n-GaAs disk Pt-ring electrode in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution as a function of rotation speed are presented in this appendix. The potential at the GaAs disk was scanned from -0.6V to $+0.2$V (Ag/AgCl) at a sweeprate of 10 mV/s. The ring electrode was held at a constant potential of -0.2V (Ag/AgCl) to monitor the flux of Fe^{+3}. The two cases presented here are for relative light intensities corresponding to 50% transmission (%T) and 100%T. The intensity of the light was varied using neutral density filters. These figures were discussed in Chapter 3, section 3.3.3.1.

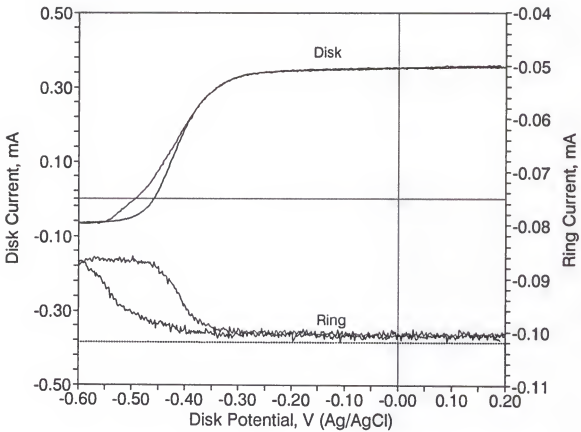


Figure C.1: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 150 RPM in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution. The relative light intensity was 50%T. The ring electrode was held at a constant potential of −0.2V (Ag/AgCl). Solid line: RRDE I-V curves. Dashed line: Unshielded ring-electrode limiting current.

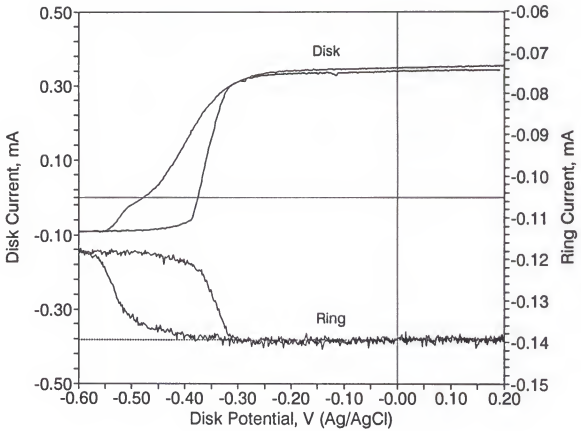


Figure C.2: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 300 RPM and 50% light transmission. Similar system description as in Figure C.1.

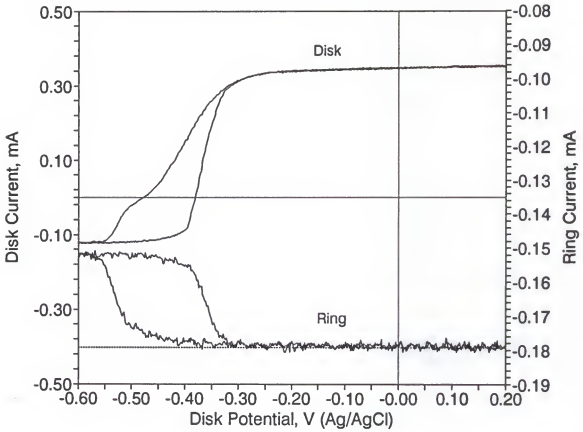


Figure C.3: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 500 RPM and 50% light transmission. Similar system description as in Figure C.1.

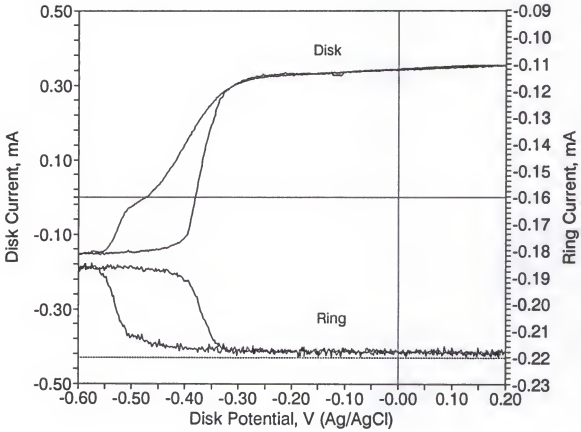


Figure C.4: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 700 RPM and 50% light transmission. Similar system description as in Figure C.1.

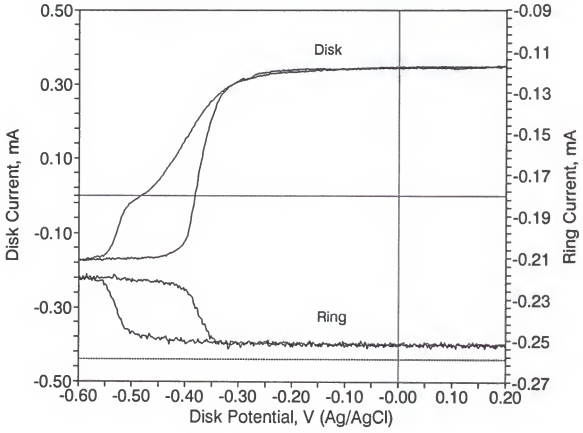


Figure C.5: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 1000 RPM and 50% light transmission. Similar system description as in Figure C.1.

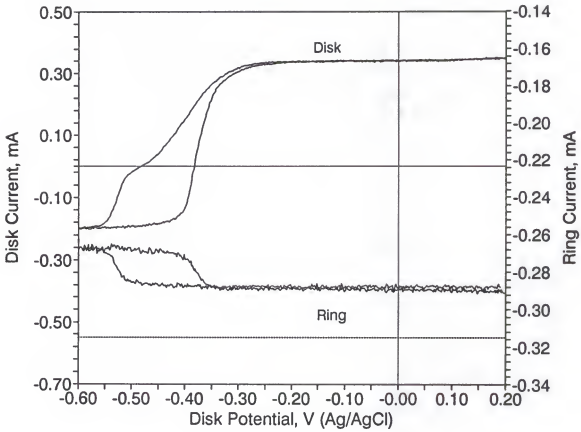


Figure C.6: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 1300 RPM and 50% light transmission. Similar system description as in Figure C.1.

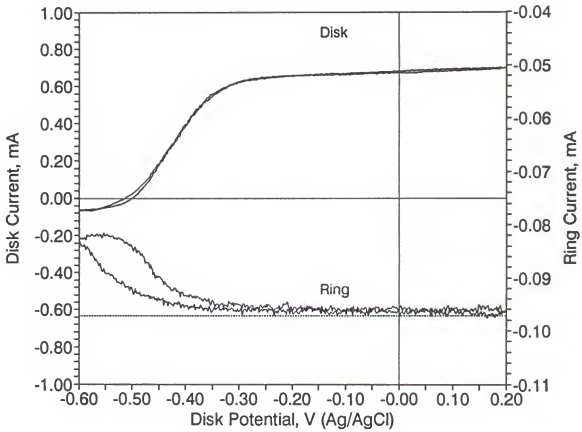


Figure C.7: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 150 RPM in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution. The relative light intensity was 100%T. The ring electrode was held at a constant potential of -0.2V (Ag/AgCl). Solid line: RRDE I-V curves. Dashed line: Unshielded ring-electrode limiting current.

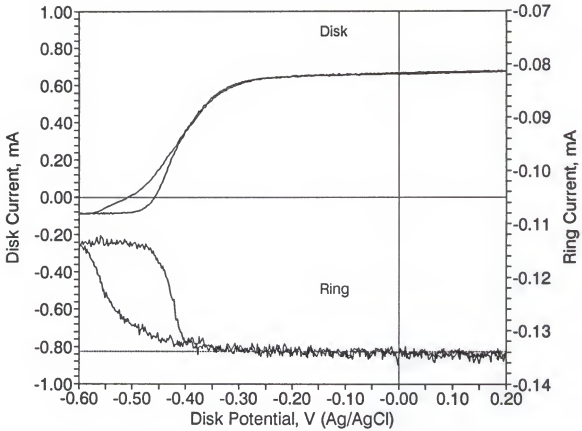


Figure C.8: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 300 RPM and 100% light transmission. Similar system description as in Figure C.7.

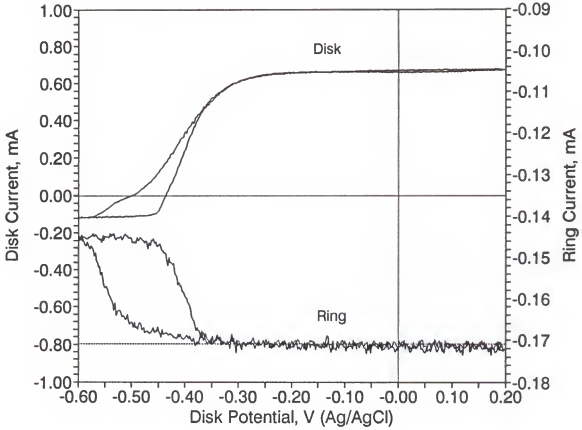


Figure C.9: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 500 RPM and 100% light transmission. Similar system description as in Figure C.7.

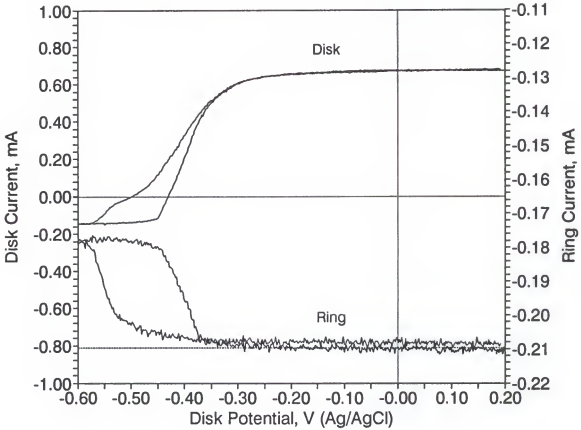


Figure C.10: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 700 RPM and 100% light transmission. Similar system description as in Figure C.7.

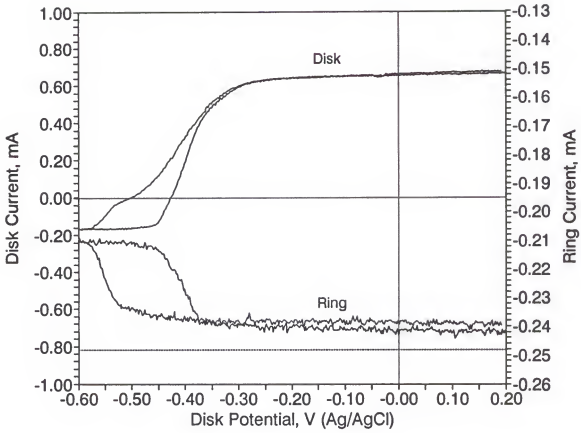


Figure C.11: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 1000 RPM and 100% light transmission. Similar system description as in Figure C.7.

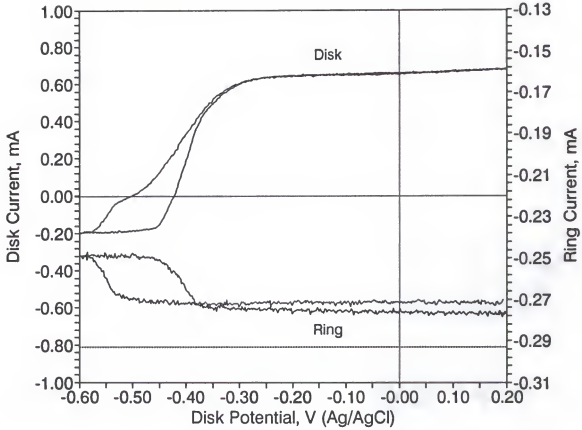


Figure C.12: Potentiodynamic scan for an illuminated n-GaAs disk Pt-ring electrode rotating at 1300 RPM and 100% light transmission. Similar system description as in Figure C.7.

Appendix D
POTENTIOSTATIC DATA FOR RRDE

Potentiostatic experiments for an illuminated n-GaAs disk Pt-ring electrode in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution as a function of applied potential are presented in this appendix. These figures were discussed in Chapter 3, section 3.3.3.2. The rotation speed of the RRDE was fixed at 500 RPM and light transmission was set at 20%T. The ring electrode was held at a constant potential of -0.2V (Ag/AgCl) to monitor the flux of Fe^{+3}. At the start of the experiment, the disk potential was held at -0.6V (Ag/AgCl) for several minutes without illumination. The disk potential was then stepped to the specified potential without illumination for several more minutes. The electrode was then illuminated for the duration of the experimental run at the specified potential. The unshielded ring currents were estimated using a known value of collection efficiency. The solid lines drawn across the ring-current transient represent the calculated unshielded ring currents based on the theoretical value of collection efficiency, $N_o = 0.225$, and for $N_o = 0.18$, as measured from previous experiments.

The deconvoluted redox and GaAs decomposition reactions at the disk electrode are also illustrated in this Appendix. The partial anodic currents were estimated using a collection efficiency of 0.20. The error bars in each graph represent the uncertainty if the experimental value of N_o were between 0.18 and 0.225. Although the total disk current was nearly constant, these graphs show that the rate of the dissolution reaction decreased with time while the rate of the stabilization reaction increased.

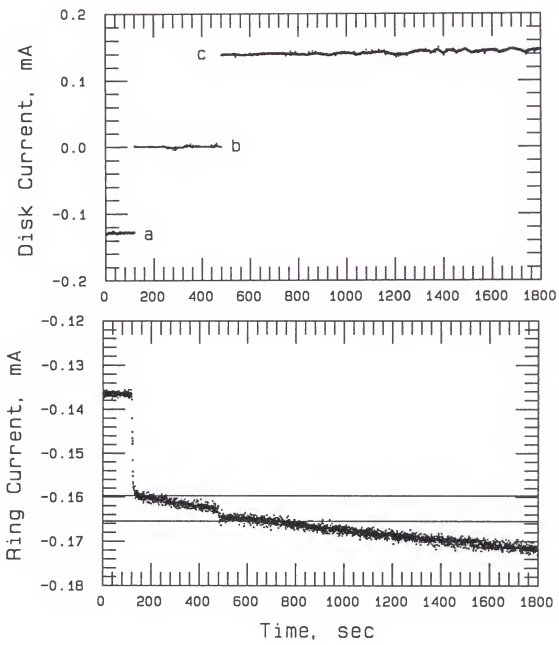


Figure D.1: Disk and ring currents for the potential step experiment (+0.1V (Ag/AgCl)) of an n-GaAs-disk/Pt-ring electrode in 0.5M Fe^{2+}, 0.25M Na_2SO_4, pH = 1.4 solution, with a rotation speed of 500 RPM. The relative light intensity was 20%T, and the potential of the ring electrode was held constant at −0.2V. a) I_d = −0.6V (Ag/AgCl) without illumination. b) I_d = +0.1V (Ag/AgCl) without illumination. c) I_d = +0.1V (Ag/AgCl) with illumination.

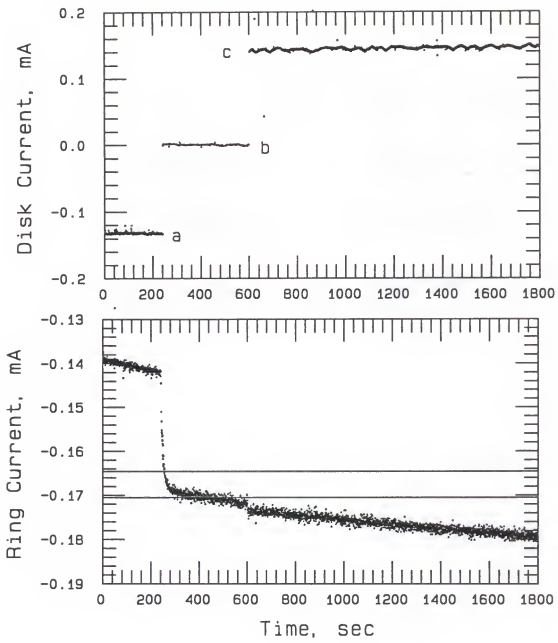


Figure D.2: Disk and ring currents for the potential step experiment (+0.0V (Ag/AgCl)) using the same experimental system described in Figure D.1. a) I_d = −0.6V (Ag/AgCl) without illumination. b) I_d = +0.0V (Ag/AgCl) without illumination. c) I_d = +0.0V (Ag/AgCl) with illumination.

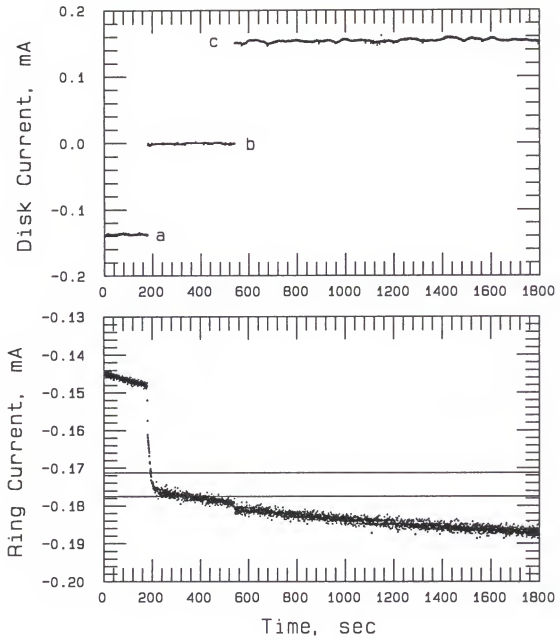


Figure D.3: Disk and ring currents for the potential step experiment (−0.1V (Ag/AgCl)) using the same experimental system described in Figure D.1. a) I_d = −0.6V (Ag/AgCl) without illumination. b) I_d = −0.1V (Ag/AgCl) without illumination. c) I_d = −0.1V (Ag/AgCl) with illumination.

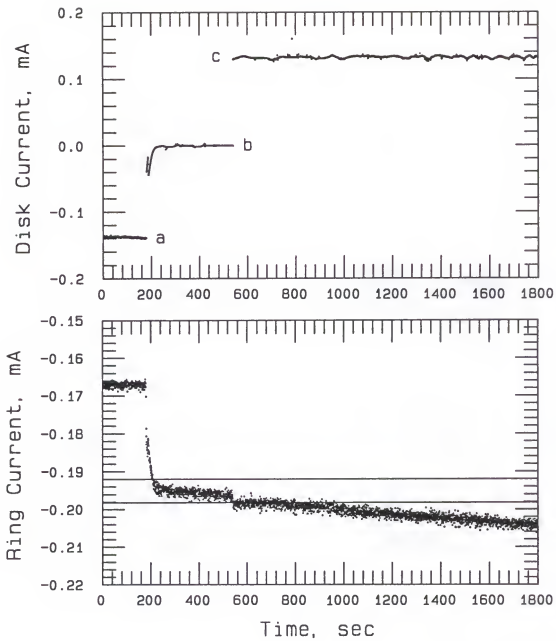


Figure D.4: Disk and ring currents for the potential step experiment (−0.2V (Ag/AgCl)) using the same experimental system described in Figure D.1. a) I_d = −0.6V (Ag/AgCl) without illumination. b) I_d = −0.2V (Ag/AgCl) without illumination. c) I_d = −0.2V (Ag/AgCl) with illumination.

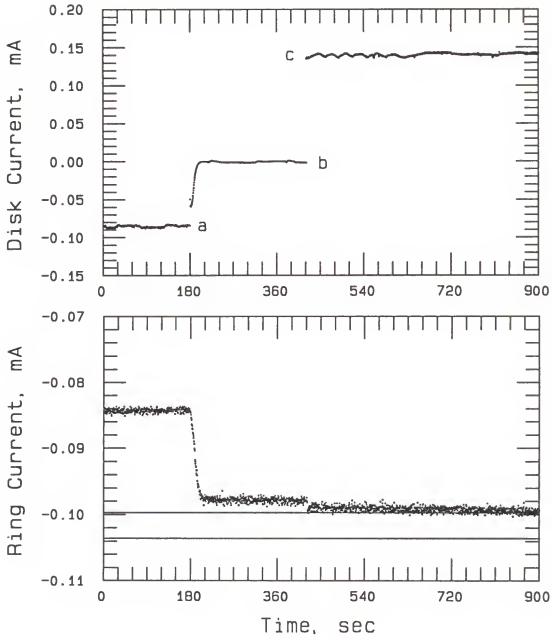


Figure D.5: Disk and ring currents for the potential step experiment (−0.25V (Ag/AgCl)) using the same experimental system described in Figure D.1. a) I_d = −0.6V (Ag/AgCl) without illumination. b) I_d = −0.25V (Ag/AgCl) without illumination. c) I_d = −0.25V (Ag/AgCl) with illumination.

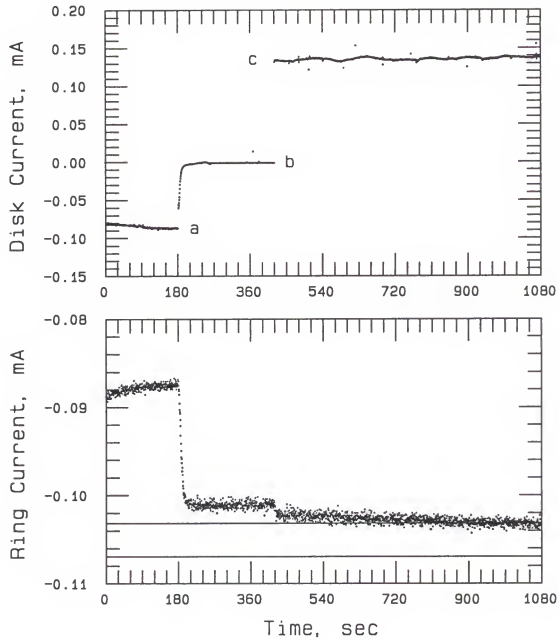


Figure D.6: Disk and ring currents for the potential step experiment (−0.3V (Ag/AgCl)) using the same experimental system described in Figure D.1. a) I_d = −0.6V (Ag/AgCl) without illumination. b) I_d = −0.3V (Ag/AgCl) without illumination. c) I_d = −0.3V (Ag/AgCl) with illumination.

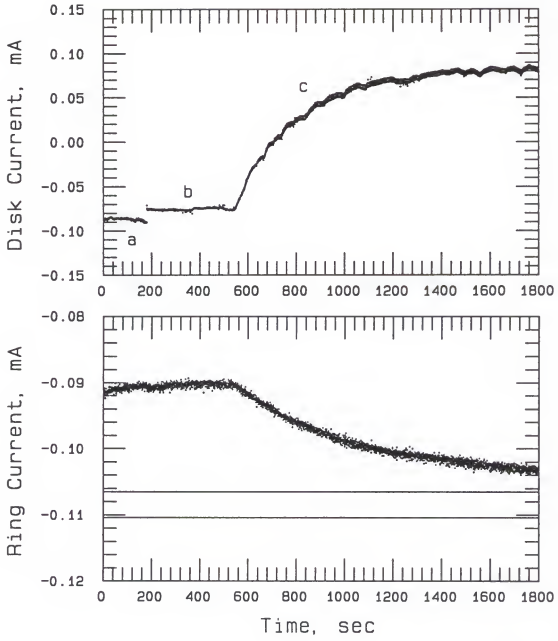


Figure D.7: Disk and ring currents for the potential step experiment (−0.35V (Ag/AgCl)) using the same experimental system described in Figure D.1. a) I_d = −0.6V (Ag/AgCl) without illumination. b) I_d = −0.35V (Ag/AgCl) without illumination. c) I_d = −0.35V (Ag/AgCl) with illumination.

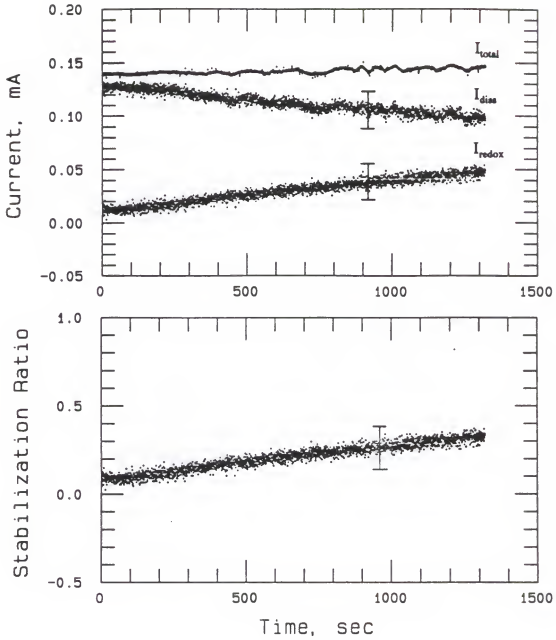


Figure D.8: Partial disk currents and stabilization ratio for the experimental system described in Figure D.1 when the disk potential is +0.1V (Ag/AgCl). a) Deconvoluted redox and GaAs decomposition currents. b) Calculated stabilization ratio, assuming $N_o = 0.20$.

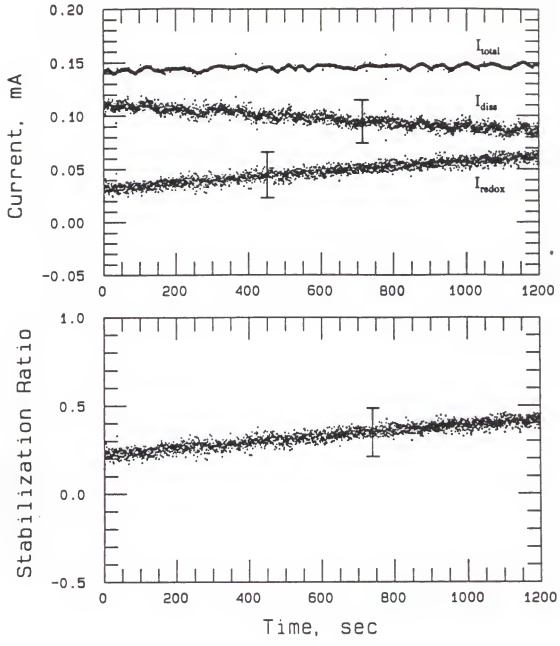


Figure D.9: Partial disk currents and stabilization ratio for the experimental system described in Figure D.1 when the disk potential is +0.0V (Ag/AgCl). a) Deconvoluted redox and GaAs decomposition currents. b) Calculated stabilization ratio, assuming $N_o = 0.20$.

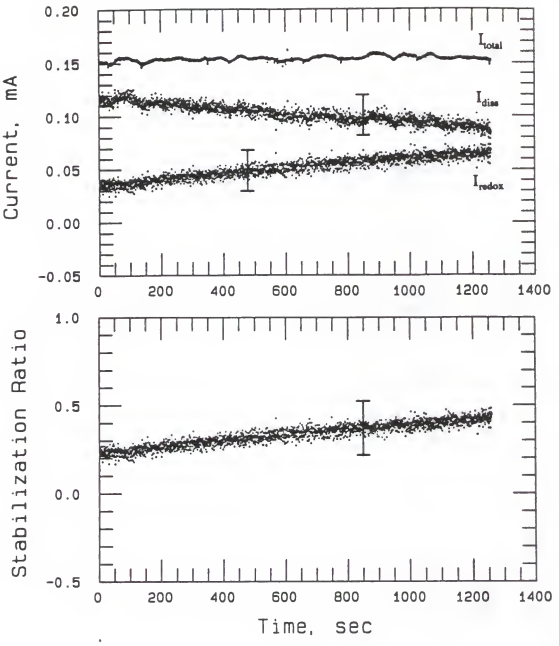


Figure D.10: Partial disk currents and stabilization ratio for the experimental system described in Figure D.1 when the disk potential is -0.1V (Ag/AgCl). a) Deconvoluted redox and GaAs decomposition currents. b) Calculated stabilization ratio, assuming $N_o = 0.20$.

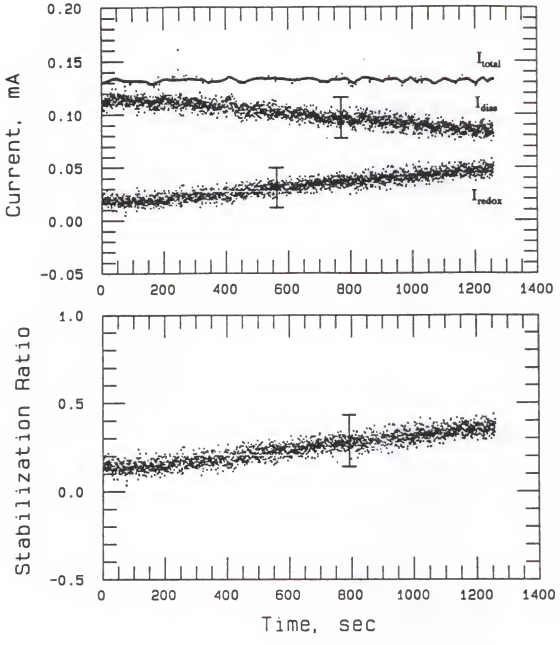


Figure D.11: Partial disk currents and stabilization ratio for the experimental system described in Figure D.1 when the disk potential is -0.2V (Ag/AgCl). a) Deconvoluted redox and GaAs decomposition currents. b) Calculated stabilization ratio, assuming $N_o = 0.20$.

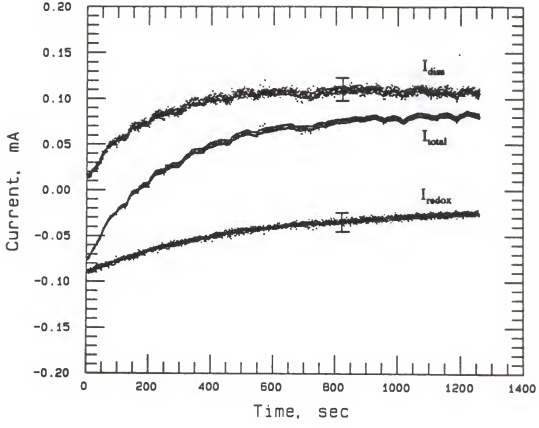


Figure D.12: Partial disk currents for the experimental system described in Figure D.1 when the disk potential is -0.35V (Ag/AgCl).

REFERENCES

[1] C. Gabrielli, *Identification of Electrochemical Processes by Frequency Response Analysis*, Solartron Instrumentation Group Monograph, The Solartron Eletronic Group Limited, Farnborough, England, 1980.

[2] J.R. Macdonald (Ed.), *Impedance Spectroscopy: Emphasizing Solid Materials and Systems*, John Wiley & Sons, New York, 1987.

[3] M. Stern and A.L. Geary, "Electrochemical Polarization: I. A Theoretical Analysis of the Shape of Polarization Curves," *Journal of the Electrochemical Society,* **104**, 56 (1957).

[4] S. Barnartt, "Two–point and Three–point Methods for the Investigation of Electrode Reaction Mechanisms," *Electrochimica Acta* **15**, 1313 (1970).

[5] J. Newman, "Ohmic Potential Measured by Interrupter Techniques," *Journal of the Electrochemical Society,* **117**, 507 (1970).

[6] S. Barnartt, "Tafel Slopes for Iron Corrosion in Acidic Solutions," *Corrosion,* **27**, 467 (1971).

[7] F. Mansfeld and K.B. Oldham, "A Modification of the Stern–Geary Linear Polarization Equation," *Corrosion Science* **11**, 787 (1971).

[8] F. Mansfeld, "Simultaneous Determination of Instantaneous Corrosion Rates and Tafel Slopes from Polarization Resistance Measurements," *Journal of the Electrochemical Society,* **120**, 515 (1973).

[9] F. Mansfeld, "Tafel Slopes and Corrosion Rates from Polarization Resistance Measurements," *Corrosion,* **29** 397 (1973).

[10] F. Mansfeld, "Some Errors in Linear Polarization Measurements and Their Correction," *Corrosion,* **30**, 92 (1974).

[11] S. M. Gershakove, L. R. Udey, and F. Mansfeld, "An Improved Method for Analysis of Polarization Resistance Data," *Corrosion,* **37**, 696 (1981).

[12] B. D. McLaughlin, "A New Approach for Determining Corrosion Currents and Tafel Slopes," *Corrosion,* **37**, 732 (1981).

[13] K. Nisancioglu, "The Error in Polarization Resistance and Capacitance Measurements Resulting from Nonuniform Ohmic Potential Drop to Flush–Mounted Probes," *Corrosion,* **43**, 258 (1987).

[14] J.M. Esteban, M. Lowry, and M.E. Orazem, "Correction of Experimental Data for the Ohmic Potential Drop Corresponding to a Secondary Current Distribution on a Disk Electrode," in *The Measurement and Correction of Electrolyte Resistance in Electrochemical Test, ASTM STP 1056,* L.L. Scribner and S.R. Taylor, Eds., p. 127, American Society for Testing and Materials, Philadelphia, 1990.

[15] J.R. Macdonald, "Impedance Spectroscopy and Its Use in Analyzing the Steady-State AC Response of Solid and Liquid Electrolytes," *Journal of Electroanalytical Chemistry,* **223**, 25 (1987).

[16] J. Ross Macdonald, *Complex Nonlinear Least Squares Immitance Fitting Program: LOMFP,* University of North Carolina, Chapel Hill, NC.

[17] H. Shih and F. Mansfeld, "A Fitting Procedure for Impedance Spectra Obtained for Cases of Localized Corrosion," *Corrosion,* **45**, 610 (1989).

[18] H. Shih and F. Mansfeld, "A Fitting Procedure for Impedance Data of Systems With Very Low Corrosion Rates," *Corrosion Science,* **29**, 1235 (1989).

[19] B.A. van Hassel, I.C. Vinke, K.J. de Vries, B.A. Boukamp, and A.J. Burggraaf, "Computer-Assisted Data Analysis in Electrochemistry," *Journal of the Electrochemical Society,* **136**, 420C (1989).

[20] M. Sluyters-Rehbach and J.H. Sluyters, "Analysis of Electrode Processes by Alternating Current and Transient Techniques: Recognition of Mechanisms," in *Transient Techniques in Corrosion Science and Engineering,* (PV89-1) W.H. Smyrl, D.D. Macdonald, and W.J. Lorenz (eds.), p. 1, The Electrochemical Society Proceedings Softbound Series, Pennington, NJ, 1989.

[21] D.B. Bonham and M.E. Orazem, "A Mathematical Model for the AC Impedance of Semiconducting Electrodes," *AICHE Journal,* **34**, 465 (1988).

[22] D.B. Bonham and M.E. Orazem, "A Mathematical Model for the Influence of Deep-Level Electronic States on Photoelectrochemical Impedance Spectroscopy: 1. Theoretical Development," To appear in *Journal of the Electrochemical Society* (1992).

[23] D.B. Bonham and M.E. Orazem, "A Mathematical Model for the Influence of Deep-Level Electronic States on Photoelectrochemical Impedance Spectroscopy: 2. Assessment of Characterization Methods Based on Mott-Schottky Theory," To appear in *Journal of the Electrochemical Society* (1992).

[24] V.A. Tyagai and G.Ya. Kolbasov, "Use of the Kramers-Kronig Expressions for the Analysis of Low-Frequency Impedance," *Élektrokhimiya,* **8**, 59 (1972).

[25] R.L. Van Meirhaeghe, E.C. Dutoit, F. Cardon, and W.P. Gomes, "On the Application of the Kramers-Kronig Relations to Problems Concerning the Frequency Dependence of Electrode Impedance," *Electrochimica Acta,* **20**, 995 (1975).

[26] D.D. Macdonald and M. Urquidi-Macdonald, "Application of Kramers-Kronig Transforms in the Analysis of Electrochemical Systems: I. Polarization Resistance," *Journal of the Electrochemical Society,* **132**, 2316 (1985).

[27] M.Urquidi-Macdonald, S. Real, and D.D. Macdonald, "Application of Kramers-Kronig Transforms in the Analysis of Electrochemical Systems: II. Transformations in the Complex Plane," *Journal of the Electrochemical Society,* **133**, 2018 (1986).

[28] M. Urquidi-Macdonald, S. Real, and D.D. Macdonald, "Applications of Kramers–Kronig Transforms in the Analysis of Electrochemical Impedance Data — III. Stability and Linearity," *Electrochimica Acta ,* **35**, 1559 (1990).

[29] C. Haili, "The Corrosion of Iron Rotating Hemispheres in 1M Sulfuric Acid: An Electrochemical Impedance Study," M.S. Thesis, University of California, Berkeley, June, 1987.

[30] A. Heller and B. Miller, "Some Recent Progress in Semiconductor–Liquid Junction Solar Cells," *Electrochimica Acta,* **25**, 29 (1980).

[31] A. Heller, B. Miller, and F.A. Thiel, "11.5% Solar Conversion Efficiency in the Photocathodically Protected p-InP/V^{3+}–V^{2+}–HCl/C Semiconductor Liquid Junction Cell," *Applied Physics Letters,*, **38**, 282 (1981).

[32] A. Heller, "Electrochemical Solar Cells," *Solar Energy,* **29**, 153 (1982).

[33] R. de L. Kronig, "On the Theory of Dispersion of X-Rays," *Journal of the Optical Society of America and Review of Scientific Instruments*, **12**, 547, (1926).

[34] R.de L. Kronig, "Dispersionstheorie im Röntgengebiet," *Physikalische Zeitschrift*, **30**, 521 (1929).

[35] H.A. Kramers, "Die Dispersion und Absorption von Röntgenstrahlen," *Physikalische Zeitschrift*, **30**, 522 (1929).

[36] B. Davies, "Integral Transforms and Their Applications," in *Applied Mathematical Series,* Volume 25 (2nd ed.), p. 101, Springer–Verlag, New York, 1985.

[37] H.W. Bode, "Network Analysis and Feedback Amplifier Design," D. Van Nostrand Company, Inc., New York, 1945.

[38] D.D. Macdonald and M. Urquidi-Macdonald, "Kramers–Kronig Transformation of Constant Phase Impedances," *Journal of the Electrochemical Society,* **137**, 515 (1990).

[39] D. Townley, "Comments: Evaluation of Kramers-Kronig Transforms for Electrochemical Impedance Data," *Journal of the Electrochemical Society,* **137**, 3305 (1990).

[40] M. Urquidi-Macdonald and D.D. Macdonald, "Comments: Evaluation of Kramers-Kronig Transforms for Electrochemical Impedance Data," *Journal of the Electrochemical Society,* **137**, 3306 (1990).

[41] M.C.H. McKubre, D.D. Macdonald, and J.R. Macdonald, "Corrosion of Materials: Kramers–Kronig Transforms," in *Impedance Spectroscopy: Emphasizing Solid Materials and Systems*, J.R. Macdonald, (Ed.), p. 274, John Wiley & Sons, New York, 1987.

[42] M. Kendig and F. Mansfeld, "Corrosion Rates From Impedance Measurements: An Improved Approach for Rapid Automatic Analysis," *Corrosion,* **39**, 466 (1983).

[43] F. Mansfeld, M.W. Kendig, W.J. Lorenz, "Corrosion Inhibition in Neutral Aerated Media," *Journal of the Electrochemical Society,* **132**, 290 (1985).

[44] H. Shih and F. Mansfeld, "Concerning the Use of the Kramers–Kronig Transforms for the Validation of Impedance Data," *Corrosion Science,* **28**, 933 (1988).

[45] H. Shih and F. Mansfeld, "Can the Kramers–Kronig Relations be Used for the Validation of Impedance Data?" in *Transient Techniques in Corrosion Science and Engineering,* (PV89-1) W.H. Smyrl, D.D. Macdonald, and W.J. Lorenz (eds.), p. 31, The Electrochemical Society Proceedings Softbound Series, Pennington, NJ, 1989.

[46] H. Shih and F. Mansfeld, "Should the Kramers–Kronig Relations be Used for the Validation of Impedance Data?" *Corrosion,* **45**, 325 (1989).

[47] F. Mansfeld and H. Shih, "Comments: Kramers–Kronig Transformation of Constant Phase Impedances," *Journal of the Electrochemical Society,* **137**, 3303 (1990).

[48] D.D. Macdonald and M. Urquidi-Macdonald, "Comment: Should the Kramers–Kronig Relations be Used for the Validation of Impedance Data?" *Corrosion,* **45**, 327 (1989).

[49] D.D. Macdonald and M. Urquidi-Macdonald, "Comments: Kramers–Kronig Transformation of Constant Phase Impedances," *Journal of the Electrochemical Society,* **137**, 3304 (1990).

[50] J.M. Esteban and M.E. Orazem, "On The Use of the Kramers–Kronig Relations to Evaluate the Consistency of Electrochemical Impedance Data," *Journal of the Electrochemical Society,* **138**, 67 (1991).

[51] M.E. Orazem, J.M.Esteban, and O.C. Moghissi, "Practical Applications of the Kramers–Kronig Relations," *Corrosion,* **47**, 248 (1991).

[52] A. Hiser, P. Gifford, M.E.Orazem, *manuscript in preparation* (1991).

[53] O.C. Moghissi, "An Electrochemical Study of Cu and Cu-Al Alloys in Saline Solutions," M.S. Thesis, University of Virginia, August, 1989.

[54] G. Bianchi and P. Longhi, "Copper in Sea–Water: Potential–pH Diagrams," *Corrosion Science,* **13**, 853 (1973).

[55] *ZPLOT: Electrochemical Impedance Software*, Scribner Associates, Inc., Charlottesville, VA, 1987.

[56] D.C. Silverman, "Rapid Corrosion Screening in Poorly Defined Systems by Electrochemical Impedance Technique," *Corrosion,* **46**, 589 (1990).

[57] M. Green, "Electrochemistry of the Semiconductor–Electrolyte Electrode. I. The Electrical Double Layer," *Journal of Chemical Physics,* **31**, 200 (1959).

[58] P.J. Boddy, "Review: The Structure of the Semiconductor–Electrolyte Interface," *Journal of Electroanalytical Chemistry,* **10**, 199 (1965).

[59] H. Gerischer and W. Mindt, "The Mechanisms of the Decomposition of Semiconductors by Electrochemical Oxidation and Reduction," *Electrochimica Acta,* **13**, 1329 (1968).

[60] H.U. Harten, "The Semiconductor/Electrolyte Interface: Potential Charges and Carriers," *Electroanalytical Chemistry,* **13**, 1255 (1968).

[61] H. Gerischer, "Semiconductor Electrochemistry," in *Physical Chemistry: An Advanced Treatise*, Volume IXA, H. Eyring and W. Sost (Eds.), p. 463, Academic Press, New York, 1970.

[62] R. Memming, "Processes at Semiconductor Electrodes," in *Comprehensive Treatise of Electrochemistry,* Volume VII, B.E. Conway *et al.* (Eds.), p. 529, Plenum Press, New York, 1983.

[63] N. S. Lewis, "Photoeffects at the Semiconductor/Liquid Interface," *Annual Reviews in Material Science,* **14**, 95 (1984).

[64] S.R. Morrison, "Study of Semiconductors Using Electrochemical Techniques," *Journal of Vacuum Science Technology,* **15**, 1417 (1978).

[65] H. Gerischer, "Über den Mechanismus der anodischen Auflösung von Galliumarsenid," *Berichte der Bunsengesellschaft für Physikalische Chemie,* **69**, 578 (1965).

[66] F. Decker, B. Pettinger, and H. Gerischer, "Hole Injection and Electroluminescence of n-GaAs in the Presence of Aqueous Redox Electrolytes," *Journal of the Electrochemical Society,* **130**, 1335 (1983).

[67] P.H.L. Notten, "The Role of Surface Charging and Potential Redistribution on the Kinetics of Hole Injection Reactions at n-GaAs," *Electroanalytical Chemistry,* **32**, 575 (1987).

[68] D. Vanmaekelbergh, J.J. Kelly, S. Lingier, and W.P. Gomes, "Electron Excitation and Chemical Steps during Anodic Decomposition of n-GaAs Electrodes: A Hole Injection Study," *Berichte der Bunsengesellschaft für Physikalische Chemie,* **92**, 1068 (1988).

[69] D. Vanmaekelbergh and J.J. Kelly, "The Anodic Decomposition Mechanism of n-GaP Electrodes: A Hole Injection Study," *Journal of the Electrochemical Society,* **136**, 108 (1989).

[70] D. Vanmaekelbergh, W.P. Gomes, and F. Cardon, "Study of Stabilization and Surface Recombination on n–GaP Photoelectrodes: Mechanisms and Interrelation," *Journal of the Electrochemical Society,* **129**, 546 (1982).

[71] A.J. Bard and Mark S. Wrighton, "Thermodynamic Potential for the Anodic Dissolution of n-type Semiconductors: A Crucial Factor Controlling Durability and Efficiency in Photoelectrochemical Cells and an Important Criterion in the Selection of New Electrode/Electrolyte Systems," *Journal of the Electrochemical Society,* **124**, 1706 (1977).

[72] H. Gerischer, "On the Stability of Semiconductor Electrodes Against Photodecomposition," *Journal of Electroanalytical Chemistry,* **82**, 133 (1977).

[73] H. Gerischer, "Electrolytic Decomposition and Photodecomposition of Compound Semiconductors in Contact With Electrolytes," *Journal of Vacuum Science Technology,* **15**, 1422 (1978).

[74] K.W. Frese, Jr., M.J. Madou, and S.R. Morrison, "Investigation of Photoelectrochemical Corrosion of Semiconductors. 1," *Journal of Physical Chemistry,* **84**, 3172 (1980).

[75] K.W. Frese, Jr., M.J. Madou, and S.R. Morrison, "Investigation of Photoelectrochemical Corrosion of Semiconductors. II. Kinetic Analysis of Corrosion–Competition Reactions on n-GaAs," *Journal of the Electrochemical Society,* **128**, 1527 (1981).

[76] S.M. Park and M.E. Barber "Thermodynamic Stabilities of Semiconductor Electrode," *Journal of Electroanalytical Chemistry,* **99**, 67 (1979).

[77] D. Laser and A.J. Bard, "Semiconductor Electrodes: VII. Digital Simulation of Charge Injection and the Establishment of the Space Charge Region in the Absence and Presence of Surface States," *Journal of the Electrochemical Society,* **123**, 1828 (1976).

[78] D. Laser, "Modes of Charge Transfer at an Illuminated Semiconductor Electrode, a Digital Simulation," *Journal of the Electrochemical Society,* **126**, 1011 (1979).

[79] M. Kazeminy and M.E. Orazem, "The Influence of Electrolytic Mass Transfer on the Performance of Photoelectrochemical Cells," *Chemical Engineering Communications,* **57**, 335 (1987).

[80] M.E. Orazem, "A Mathematical Model For the Photoelectrochemical Etching of Semiconductors," in *Electrochemical Engineering Applications,* R.E. White, R.F. Savinell, and A. Schneider (Eds,). p. 25, American Institute of Chemical Engineers, New York, AICHE Symposium Series, **83**, No. 254, 1987.

[81] R.L. Van Meirhaeghe, F. Cardon, and W.P. Gomes, "On the Influence of the Surface Pretreatment on the Photocurrent–Voltage Characteristics and the Spectral Response of the n-GaAs/Electrolyte Interface," *Berichte der Bunsengesellschaft für Physikalische Chemie,* **83**, 236 (1979).

[82] B. Miller, "Charge Transfer and Corrosion Processes at III-V Semiconductor/Electrolyte Interfaces," *Journal of Electroanalytical Chemistry,* **168**, 91 (1984).

[83] D. Vanmaekelbergh and W.P. Gomes, "Relationship between Stabilization and Photocurrent at n-Type Semiconductor Electrodes on the Basis of a Model Including Surface Recombination through Decomposition Intermediates," *Journal of the Chemical Society, Faraday Transactions 1,"* **79**, 1391 (1983).

[84] D. Vanmaekelbergh, W. Rigole, W.P. Gomes, and F. Cardon, "Role of Decomposition-Induced and Other Surface States in the Photoelectrochemical Kinetics of n-GaP/Fe^{2+}," *Journal of the Chemical Society, Faraday Transactions 1,"* **79**, 2813 (1983).

[85] D. Vanmaekelbergh, W.P. Gomes, and F. Cardon, "Studies on the n-GaAs Photoanode in Aqueous Electrolytes. 1. Behaviour of the Photocurrent in the Presence of a Stabilizing Agent," *Berichte der Bunsengesellschaft für Physikalische Chemie,* **89**, 987 (1985).

[86] K.C. Chang, A. Heller, B. Schwartz, and S. Menezes, and B. Miller, "Stable Semiconductor Liquid Junction Cell with 9 Percent Solar–to–Electrical Conversion Efficiency," *Science,* **196**, 1097 (1977).

[87] B.A. Parkinson, A. Heller, and B. Miller, "Enhanced Photoelectrochemical Solar–Energy Conversion by Gallium Arsenide Surface Modification," *Applied Physics Letters,* **33**, 521 (1978).

[88] A. Heller, H.J. Lewerenz, and B. Miller, "Combined Ruthenium Lead Surface Treatment of Gallium Arsenide Photoanodes," *Berichte der Bunsengesellschaft für Physikalische Chemie,* **84**, 592 (1980).

[89] A. Heller, H.J. Leamy, B. Miller, and W.D. Johnston, Jr., "Chemical Passivation of Carrier Recombination at Acid Interfaces and Grain Boundaries of p-InP," *Journal of Physical Chemistry,* **87**, 3239 (1983).

[90] S. Menezes, A. Heller, and B. Miller, "Metal Filmed–Semiconductor Photoelectrochemical Cells," *Journal of the Electrochemical Society,* **127**, 1268 (1980).

[91] F. Cardon, W.P. Gomes, F.V. Kerchove, D. Van Maekelbergh, and F. Van Overmeire, "On the Kinetics of Semiconductor–Electrode Stabilization," *Faraday Discussion of the Chemical Society ,* **70**, 153 (1980).

[92] D. Vanmaekelbergh, W.P. Gomes, and F. Cardon, "Studies on the n-GaAs Photoanode in Aqueous Electrolytes. 2. Differential Capacitance Behaviour," *Berichte der Bunsengesellschaft für Physikalische Chemie,* **89**, 994 (1985).

[93] D. Vanmaekelbergh, W.P. Gomes, and F. Cardon, "Studies on the n-GaAs Photoanode in Aqueous Electrolytes. 3. Recombination Resistance," *Berichte der Bunsengesellschaft für Physikalische Chemie,* **90**, 431 (1986).

[94] V.G. Levich, *Physicochemical Hydrodynamics*, Prentice–Hall, Inc., Englewood Cliffs, NJ (1962).

[95] A.C. Riddiford, "The Rotating Disk System", in *Advances in Electrochemistry and Electrochemical Engineering,* Paul Delahay, (Ed.), **4**, 47 (1965).

[96] J. Newman, "Current Distribution on a Rotating Disk Below the Limiting Current," *Journal of the Electrochemical Society,* **113**, 1235 (1966).

[97] W.H. Smyrl and J.S. Newman, "Ring–Disk and Sectioned Disk Electrodes," *Journal of the Electrochemical Society,* **119**, 212 (1972).

[98] B. Miller and M.I. Bellavance, "Measurement of Current and Potential Distribution at Rotating–Disk Electrodes," *Journal of the Electrochemical Society,* **120**, 42 (1973).

[99] W.J. Albery, "Ring–Disk Electrodes. Part 1. A New Approach to Theory," *Transactions of the Faraday Society,* **62**, 1915 (1966).

[100] W.J. Albery and S. Bruckenstein, "Ring–Disk Electrodes. Part 2. Theoretical and Experimental Collection Efficiencies," *Transactions of the Faraday Society,* **62**, 1920 (1966).

[101] W.J. Albery, S. Bruckenstein, and D.T. Napp, "Ring–Disk Electrodes. Part 3. Current–Voltage Curves at the Ring Electrode with Simultaneous Currents at the Disc Electrode," *Transactions of the Faraday Society,* **62**, 1932 (1966).

[102] W.J. Albery, S. Bruckenstein, and D.C. Johnson, "Ring–Disk Electrodes. Part 4. Diffusion Layer Titration Curves," *Transactions of the Faraday Society,* **62**, 1938 (1966).

[103] W.J. Albery and S. Bruckenstein, "Ring–Disk Electrodes. Part 5. First–Order Kinetic Collection Efficiencies at the Ring Electrode," *Transactions of the Faraday Society,* **62**, 1946 (1966).

[104] W.J. Albery and S. Bruckenstein, "Ring–Disk Electrodes. Part 6. Second–Order Reactions," *Transactions of the Faraday Society,* **62**, 2584 (1966).

[105] W.J. Albery and S. Bruckenstein, "Ring–Disk Electrodes. Part 7. Homogeneous and Heterogeneous Kinetics," *Transactions of the Faraday Society,* **62**, 2596 (1966).

[106] W.J. Albery, "Ring–Disk Electrodes. Part 8. Transient Currents and First–Order Kinetics," *Transactions of the Faraday Society,* **63**, 1771 (1967).

[107] W.J. Albery, M.L. Hitchman, and J. Ulstrup, "Ring–Disk Electrodes. Part 9. Application to First–Order Kinetics," *Transactions of the Faraday Society,* **64**, 2831 (1968).

[108] W.J. Albery, M.L. Hitchman, and J. Ulstrup, "Ring–Disk Electrodes. Part 10. Application to Second–Order Kinetics," *Transactions of the Faraday Society,* **65**, 1101 (1969).

[109] W.J. Albery, "Ring–Disk Electrodes. Part 11. General Theory of Transient Currents," *Transactions of the Faraday Society,* **67**, 153 (1971).

[110] W.J. Albery, J.S. Drury, and M.L. Hitchman, "Ring–Disk Electrodes. Part 12. Application to Ring Current Transients," *Transactions of the Faraday Society,* **67**, 161 (1971).

[111] W.J. Albery, J.S. Drury, and M.L. Hitchman, "Ring–Disk Electrodes. Part 13. The Laplace Transformation of Transients," *Transactions of the Faraday Society,* **67**, 166 (1971).

[112] W.J. Albery, J.S. Drury, and M.L. Hitchman, "Ring–Disk Electrodes. Part 14. Kinetic and Transient Parameters," *Transactions of the Faraday Society,* **67**, 2162 (1971).

[113] W.J. Albery, J.S. Drury, and A.P. Hutchinson, "Ring–Disk Electrodes. Part 15. Alternating Current Measurements," *Transactions of the Faraday Society,* **67**, 2414 (1971).

[114] W.J. Albery and M.L. Hitchman, *Ring–Disc Electrodes,* Oxford University Press, London (1971).

[115] J.S. Newman, *Electrochemical Systems*, Prentice–Hall, Inc., Englewood Cliffs, NJ (1973).

[116] D.P. Gregory and A.C. Riddiford, "Transport to the Surface of a Rotating Disk", *Journal of the Chemical Society,* 3756 (1956).

[117] D.T. Napp, D.C. Johnson, and S. Bruckenstein, "Simultaneous and Independent Potentiostatic Control of Two Indicator Electrodes: Application to the Copper(II)/Copper(I)/Copper System in 0.5M KCl at the Rotating Ring–Disc Electrode," *Analytical Chemistry,* **39**, 481 (1967).

[118] W.J. Albery and J. Ulstrup, "The Current Distribution on a Rotating Disk Electrode," *Electrochimica Acta,* **13**, 281 (1968).

[119] *Pascal, C, and Fortran: Callable Drivers for Metrabytes DAS-16, DAS-16F, DAS-16G, and uCDAS-16G*, Metrabyte Corporation, Taunton, MA (1989).

[120] *Hewlett–Packard 7090A Measurement Plotting System Interfacing and Programming Manual: Using HP–RL and HP–GL Instructions*, Hewlett–Packard Company, San Diego, CA (1984).

[121] *Model 273 Potentiostat/Galvanostat: Operating Manual*, EG&G Princeton Applied Research, Princeton, NJ (1985).

[122] S.L. Gordon, J.S. Newman, and C.W. Tobias, "The Role of Ionic Migration in Electrolytic Mass Transport; Diffusivities of $[Fe(CN)_6]^{3-}$ and $[Fe(CN)_6]^{4-}$ in KOH and NaOH Solutions," *Berichte der Bunsengesellschaft für Physikalische Chemie,* **70**, 414 (1966).

[123] R.L. Burden and J.D. Faires, *Numerical Analysis*, (4nd ed.), Prindle, Weber & Schmidt, Boston, MA, 1989.

[124] M. Abromowitz and I.A. Stegun, eds., *Handbook of Mathematical Functions,* 9th printing, Dover Publications, Inc., New York, NY.

[125] F.B. Growcock and R.J. Jasinski, "Time–Resolved Impedance Spectroscopy of Mild Steel in Concentrated Acid," *Journal of the Electrochemical Society,* **136**, 2310, (1989).

BIOGRAPHICAL SKETCH

The author, Jose Matthew Rodriguez Esteban, was born on April 22, 1960, in Manila, Philippines, to Virgilio K. and Concepcion R. Esteban. His exposure to the chemical engineering profession began at an early age when his father was assigned to the Shell Oil Refinery in Tabangao, Batangas, in 1964. The family relocated to Manila in 1968.

In April, 1981, the author decided to join the religious order of the Brothers of the Christian Schools (FSC), the administrators of the De La Salle Schools he had been attending since elementary. He had the opportunity to teach in the elementary and high school departments of La Salle Greenhills, Mandaluyong. He received his B.S. Chemical Engineering degree from De La Salle University, Manila in April, 1982. During the course of the Novitiate year, Matthew decided that the religious life was not his vocation, and he requested leave of the Order on November, 1984. After leaving the Novitiate, he worked for three months with International Good Housekeeping, Inc., and completed a feasibility report on the production and marketing of rice bran oil.

Matthew came to the United States of America in May, 1985 to enroll at the University of Virginia, Charlottesville, VA. He later joined Professor Mark E. Orazem's electrochemical engineering research group and completed his research on the use of the impinging–jet electrode system to evaluate the performance of organic corrosion inhibitors under a high–shear environment. The author received his M.S. degree in chemical engineering on August, 1987. Matthew transferred to this department when Professor Orazem was recruited to the University of Florida in August, 1988. He hopes to receive his Ph.D. degree in chemical engineering on December, 1991.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Mark E. Orazem, Chairman
Associate Professor of Chemical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Timothy J. Anderson
Professor of Chemical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Hong H. Lee
Professor of Chemical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Sheng S. Li
Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Chang W. Park
Assistant Professor of Chemical Engineering

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December 1991

for Winfred M. Phillips
Dean, College of Engineering

Madelyn M. Lockhart
Dean, Graduate School